Agnel Charities

Fr. C. Rodrigues Institute of Technology

Sector 9A, Vashi, Navi Mumbai, 400703, Maharashtra, India

www.fcrit.ac.in

An Autonomous Institute Affiliated to the University of Mumbai

Department of Computer Science and Engineering

Curriculum Structure and Syllabi of Second Year B. Tech.

Prepared by : Board of Studies for Department of Computer Science and Engineering

Revision: N2025

Effective from :2026-27 (N2025)

Curriculum Structure – SY Semester-III

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		L	P	T	L	P	T	Total
CSPCC301	Engineering Mathematics-III	4	--	--	4	--	--	4
CSPCC302	Discrete Structures	3	--	1	3	--	1	4
CSPCC303	Data Structures	3	--	--	3	--	--	3
CSPCC304	Database Management System	3	--	--	3	--	--	3
XXMDM301Y	--	3	--	--	3	--	--	3
CSLBC301	Data Structure Laboratory	--	2	--	--	1	--	1
CSLBC302	SQL Laboratory	--	2	--	--	1	--	1
CSSBL301	Python Laboratory	--	4	--	--	2	--	2
CSMNP301	Mini Project-1A	--	3	--	--	1	--	1
HSS301	Product Design	2	--	--	2	--	--	2
	Total	**18**	**11**	**1**	**18**	**5**	**1**	**24**

NOTE:Four theory courses(Three 3-credit and one 4-credit) and one laboratory course (1-credit) offered by other department has to be taken by CSE students, to complete 14-credit requirement for MDM.

Examination Scheme – SY Semester-III

Course Code	Course Name	Examination Scheme					Total
		In-Semester Assessment$		End Sem Exam (ESE)	Exam Duration for Theory (in Hrs)		
		Continuous Assessment	Mid-Sem Exam		Mid-Sem	End-Sem	
CSPCC301	Engineering Mathematics-III	20	30	50	1.5	2	100
CSPCC302	Discrete Structures	20+25@	30	50	1.5	2	125
CSPCC303	Data Structures	20	30	50	1.5	2	100
CSPCC304	Database Management System	20	30	50	1.5	2	100
XXMDM301Y	--	20	30	50	1.5	2	100
CSLBC301	Data Structure Laboratory	25	--	25	--	--	50
CSLBC302	SQL Laboratory	25	--	25	--	--	50
CSSBL301	Python Laboratory	50	--	50	--	--	100
CSMNP301	Mini Project-1A	50	--	--	--	--	50
HSS301	Product Design	50	--	--	--	--	50
	Total	**325**	**150**	**350**	--	--	**825**

$Please refer to the Curriculum Book of department for guidelines on in-semester assessments for both theory and laboratory courses.

@For continuous assessment of tutorials.

Curriculum Structure – SY Semester-IV

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		L	P	T	L	P	T	Total
CSPCC405	Engineering Mathematics-IV	4	--	--	4	--	--	4
CSPCC406	Design & Analysis of Algorithm	3		--	3	--	--	3
CSPCC407	Operating System	3	--	--	3	--	--	3
CSPCC408	Software Engineering	3	--	--	3	--	--	3
XXMDM402Y	--	3	--	--	3	--	--	3
CSLBC403	Design & Analysis of Algorithm Laboratory	--	2	--	--	1	--	1
CSLBC404	Linux Laboratory	--	2	--	--	1	--	1
CSLBC405	Software Development Laboratory	--	2	--	--	1	--	1
CSSBL402	Full stack development Laboratory	--	4	--	--	2	--	2
CSMNP402	Mini Project-1B	--	3	--	--	1	--	1
VEC402	Environment and Sustainability	2	--	--	2	--	--	2
	Total	**18**	**13**	**--**	**18**	**6**	**--**	**24**

Examination Scheme – SY Semester-IV

Course Code	Course Name	Examination Scheme					Total
		In-Semester Assessment$		End Sem Exam (ESE)	Exam Duration for Theory (in Hrs)		
		Continuous Assessment	Mid-Sem Exam		Mid-Sem	End-Sem	
CSPCC405	Engineering Mathematics-IV	20	30	50	1.5	2	100
CSPCC406	Design & Analysis of Algorithm	20	30	50	1.5	2	100
CSPCC407	Operating System	20	30	50	1.5	2	100
CSPCC408	Software Engineering	20	30	50	1.5	2	100
XXMDM402Y	--	20	30	50	1.5	2	100
CSLBC403	Design & Analysis of Algorithm Laboratory	25	--	25	--	--	50
CSLBC404	Linux Laboratory	25	--	25	--	--	50
CSLBC405	Software Development Laboratory	25	--	25	--	--	50
CSSBL402	Full stack development Laboratory	50	--	50	--	--	100
CSMNP402	Mini Project-1B	50	--	50	--	--	100
VEC402	Environment and Sustainability	50	--	--	--	--	50
	Total	**325**	**150**	**425**	--	--	**900**

$Please refer to the Curriculum Book of department for guidelines on in-semester assessments for both theory and laboratory courses.

@For continuous assessment of tutorials.

Course Type	Course Code	Course Name	Credits
PCC	CSPCC301	ENGINEERING MATHEMATICS- III	04

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Exam (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite :

1. BSC101 Engineering Mathematics-I

Program Outcomes addressed :

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO5: Engineering tool usage
4. PO8: Individual and team work
5. PO11: Life Long Learning

Course Objectives :

1. To provide the basic knowledge on the concepts of Mathematics in the field of Engineering.
2. To build a foundation to the methodology necessary for solving problems by applying the knowledge of Mathematics to the field of Engineering.

Module	Details	Hrs
00.	**Course Introduction** Engineering Mathematics III is often a foundational course designed to provide students with the mathematical tools and concepts essential for various engineering disciplines. Engineering Mathematics III has many applications in Information Technology such as 1) Application of Vector space in Deep Learning, dimension reduction and Neural Networks 2) Application of Linear Mapping in Deep learning and Machine Learning. 3) Application of in Number theory in Cryptography. **MATLAB Introduction** As the course aims to bridge the gap between conceptual understanding and computational practice, MATLAB is introduced as a supporting tool. Fundamental operations and basic usage of MATLAB are taught prior to the formal commencement of the curriculum.	01 01
01.	**Number Theory**	08-10

	Learning Objective/s: • *Analysis of linear congruence and systems of simultaneous congruence to identify the appropriate number theory theorems* • *Application of these theorems to solve linear congruence and simultaneous systems of congruence, where solutions exist.*	
	Contents: Primes, Modular Arithmetic, Divisibility and Euclidean Algorithm, Congruence, Residues, Computing Inverse in congruence, Euler's and Fermat Little Theorem, Chinese Remainder Theorem.	
	Self-Learning Topics: *Prime Number Theorem and Quadratic Residues*	
	Learning Outcomes : *A learner will be able to* *LO 1.1: Apply basic addition and division techniques to compute the residues and/or the modulo groups. (P.I.-1.1.1)* *LO 1.2: Identify whether modular inverse or Euclidean algorithm is to be applied to the given Linear Diophantine Equation. (P.I.- 2.2.3)* *LO 1.3: Evaluate the information presented in a given problem and thereby identify whether Euler's theorem, Fermat's Theorem or Chinese Remainder Theorem is to be used. (P.I.-2.1.1)* *LO 1.4: Apply fundamental techniques of finding remainders to solve simultaneous linear congruence's using Chinese Remainder theorem. (P.I.-1.3.1)* *LO 1.5: Analyze information present, in the given real word mathematical problem and apply the concept of Euler's theorem, Fermat's Theorem and/or Chinese Remainder Theorem to determine the solution, thereby maintaining proficiency with cryptographic techniques. (P.I.-2.1.3, P.I.-11.1.1)* *LO 1.6: Work together as a team in analyzing the information present in a given case study and apply the concept of Modular congruence to not only determine mathematical solution via MATLAB but also maintaining a conceptual and computational balance. (P.I.-5.2.2, P.I.-8.3.1, P.I.-11.3.1)*	
02.	**Vector Space** ***Learning Objective/s:*** • *Identifying if a given set is a vector space or not.* • *Analysis of vector space, its subspace, basis, dimension and dependency.* • *Application of these concepts to determine redundant vectors.*	**08-10**
	Contents: Vector Spaces, Examples of vector spaces, Subspace, Linear combinations, Spanning Set, Linear Independence and Dependence. Basis and Dimension of Vector Space and Vector Subspace.	
	Self-Learning Topics: *Independence and Dependence of functions*	
	Learning Outcomes: ***A learner will be able to***	

	LO 2.1: Apply the axioms of closure, addition and scalar multiplication and prove that the given set of vectors is a Vector Space (P.I.- 1.1.1) *LO 2.2: Identify whether the axioms of a vector subspace hold for a given space by using the appropriate mathematical algorithmic knowledge. (P.I.- 2.1.3)* *LO 2.3: Applying the fundamentals of row reductions for linear independency in order to determine the span, basis and dimension of a vector space. (P.I.- 1.3.1)* *LO 2.4: Evaluate the statements given in a problem and thereby identify whether linear dependence or independence of a set is to be considered. (P.I.- 2.1.1)* *LO 2.5: Analyze information present, in the given real word mathematical problem and apply the concept of dependent vectors to find feasible and eliminate redundant vectors useful in Data Compression, Neural Network embedding's, etc. (P.I.- 2.1.1, P.I.-11.3.2)* *LO 2.6: Work together as a team in identifying the appropriate mathematical conditions of closure and scalar multiplication required for a given space to be a Vector Subspace and analyze individually via examples/case studies how subspaces represent meaningful and reduced solution spaces within large datasets. (P.I.- 8.2.1, P.I.-11.1.1)*	
03.	**Linear Mappings** ***Learning Objective/s:*** • *Analysis of a linear map and its rank, nullity and non-singular mapping.* • *Application of these concepts to determine isomorphism between any two vector spaces.* **Contents:** Mappings, Function, Linear mapping (Linear transformations), Kernel and Image of a linear mapping, Rank and Nullity, Singular and non-singular mapping, Isomorphism. ***Self-Learning Topics:*** *Rank Nullity Theorem.* ***Learning Outcomes:*** *A learner will be able to* *LO 3.1: Apply the properties of vector addition and scalar multiplication to examine whether the mapping is a linear mapping. (P.I.- 1.1.1)* *LO 3.2: Apply laws of matrix transformations to obtain the Rank and Nullity of a given mapping. (P.I.-1.2.1)* *LO 3.3: Examine whether the given mapping is an Isomorphism using the results of non-singularity and linear mapping. (P.I.-2.1.3)* *LO 3.4: Identify whether rows and columns are to be formed for a given linear transformation based on whether Image or Kernel is to be found or to determine its non-singularity. (P.I.-2.2.3)*	**07-09**
04.	**Linear Mappings and Matrices** ***Learning Objective/s:*** • *Analysis of the concept of matrix representations and change of basis.* • *Application of these concepts in Similarity and in the context of Image Processing.* **Contents:**	**08-10**

	Matrix Representations of a linear operator, Change of Basis, Similarity. Mathematical Examples on Image Processing: Image Contrast Enhancement.	
	Self-Learning Topics: *Composition mapping*	
	Learning Outcomes: *A learner will be able to* *LO 4.1: Apply fundamentals of addition and scalar multiplication of mapping to represent it in its matrix form. (P.I.-1.3.1)* *LO 4.2: Apply rules of addition, multiplication ad inverses of matrices to check the similarity of two mappings. (P.I.-1.1.1)* *LO 4.3: Identify the domain and codomain given and determine the change of Basis Matrix from one coordinate system to the alternate coordinate system and thereby develop the rationale for identifying original and transformed data representations. (P.I.-2.1.3)* *LO 4.4: Identify the existing method of finding inverse from the change of basis matrix using similarity of matrices. (P.I.-2.2.3)* *LO 4.5: Work together as a team in identifying relevant examples from Image Processing which involves the concept of matrix representation, change of basis and thereby, determine and present the findings of the team using Similarity if the Enhanced image and the original image is the same. (P.I.-5.1.1, P.I.-8.3.1, P.I.-11.3.2)*	
05.	**Inner Product spaces, Orthogonality** ***Learning Objective/s:*** • *Analysis of various Inner Product Spaces, Orthonormal and Orthogonal vectors.* • *Application of these concepts for Gram Schmidt Orthogonalization Process.*	**07-09**
	Contents: Inner product spaces, Examples of Inner product spaces, Cauchy-Schwartz Inequality: Euclidean, Matrix and Function Space, Orthogonal sets and Basis, Orthonormal sets and Basis. Gram-Schmidt orthogonalization process: Euclidean, Matrix and Function Space	
	Self-Learning Topics: *Complex Inner Products Spaces*	
	Learning Outcomes: *A learner will be able to* *LO 5.1: Identify whether the given vector space is an Inner Product Space using linearity, Symmetric and Positive Definite Property of an Inner Product Space. (P.I.-2.1.3)* *LO 5.2: Prove Cauchy's Schwartz inequality using definitions describing the space. (P.I.-1.1.1)* *LO 5.3: Identify whether the set of vectors form an orthogonal or orthonormal basis using the standard operations of Inner Product Space. (P.I. -2.2.3)* *LO 5.4: Apply laws of dot product to determine the orthogonal and orthonormal basis using Gram Schmidt Process. (P.I. -1.2.1)*	

06.	**Numerical Methods** ***Learning Objective/s:*** • *Analysis of appropriate numerical methods for solving algebraic equations, transcendental equations, and systems of simultaneous equations* • *Application of these numerical methods to obtain approximate solutions, where they exist.*	**07-09**
	Contents: Algebraic and Transcendental Equations. Newton Raphson method, Regula –Falsi Method, Solution of system of linear algebraic equations: Gauss Jacobi Iteration Method, Gauss Seidel Iteration Method.	
	Self-Learning Topics: *Bisection Method, Gauss Elimination Method.*	
	Learning Outcomes: *A learner will be able to* *LO 6.1: Apply Newton Raphson method or Regula Falsi method to solve the transcendental equation or algebraic equations. (P.I.-1.1.1)* *LO 6.2: Apply the fundamental arithmetic techniques to solve the system of equations using Gauss Siedel and/or Gauss Jacobi methods. (P.I-1.3.1)* *LO 6.3: Identify the appropriate numerical method which can be applied to solve the given equation or system of equations. (P.I- 2.1.3)* *LO 6.4: Analyze and solve any given word problem and compare the results obtained by Gauss Siedel and Gauss Jordon methods and thereby make conclusions about its rate of convergence using MATLAB thereby maintaining proficiency with OPTIMZATION techniques (P.I.-2.2.4, P.I.-5.1.2, P.I.11.1.1)* *LO 6.5: Work together as a team and source information from case studies/research papers based on optimization techniques and present the mathematical solution maintaining a conceptual and computational balance. (P.I.-8.3.1, P.I.-11.3.1)*	
	Course Conclusion Engineering Mathematics provides the language and framework through which engineers model, analyse, and optimize systems.	**01**
Total		**60**

Performance Indicators:

P.I. No.	P.I. Statement
1.1.1	Apply the knowledge of discrete structures, linear algebra, statistics and numerical techniques to solve problems
1.2.1	Apply laws of natural science to an engineering problem
1.3.1	Apply engineering fundamentals
2.1.1	Evaluate problem statements and identifies objectives
2.1.3	Identify mathematical algorithmic knowledge that applies to a given problem
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions
2.2.4	Compare and contrast alternative solution/methods to select the best methods
5.1.1	Identify modern engineering tools, techniques and resources for engineering activities

5.2.2 Demonstrate proficiency in using discipline specific tools
8.2.1 Demonstrate effective communication, problem solving, conflict resolution and leadership skills
8.3.1 Present results as a team, with smooth integration of contributions from all individual efforts
11.1.1 Describe the rationale for requirement for continuing professional development
11.3.1 Source and comprehend technical literature and other credible sources of information
11.3.2 Analyse sourced technical and popular information for feasibility, viability, sustainability, etc.

Course Outcomes:

1. Identify and apply appropriate numerical methods to solve System of equations. *(LO 1.1, LO 1.2, LO 1.3, LO 1.4, LO 1.5)*
2. Apply the fundamentals of Vector Space to analyse its Basis, Dimension and Dependency. *(LO 2.1, LO 2.2, LO 2.3, LO 2.4, LO 2.5)*
3. Apply the fundamentals of Linear Mapping in analysing Isomorphism mapping and Similarity mapping. (*LO 3.1, LO 3.2, LO 3.3, LO 3.4, LO 4.1, LO 4.2, LO 4.3, LO 4.4)*
4. Analyse the Inner Product Space and apply the properties to determine the orthogonal basis. *(LO 5.1, LO 5.2, LO 5.3, LO 5.4)*
5. Identify and apply appropriate numerical methods to solve the System of equations. *(LO 6.1, LO 6.2, LO 6.3)*
6. Analyse and apply mathematical tools acquired through the course and thereby, exhibit the knowledge through written solutions or presentations as a team and/or individual activities. *(LO 1.6, LO 2.6, LO 4.5, LO 6.4, LO 6.5)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC301.1	3	3	-	-	-	-	-	-	-	-	3
CSPCC301.2	3	3	-	-	-	-	-	-	-	-	3
CSPCC301.3	3	3	-	-	-	-	-	-	-	-	-
CSPCC301.4	3	3	-	-	-	-	-	-	-	-	-
CSPCC301.5	3	3	-	-	-	-	-	-	-	-	-
CSPCC301.6	-	-	-	-	3	-	-	3	-	-	3
Average	**3**	**3**	**-**	**-**	**3**	**-**	**-**	**3**	**-**	**-**	**3**

Text Books :

1. Seymour Lipschutz “Beginning Linear Algebra” Schaum’s outline series, Mc-Graw Hill Publication
2. Cryptograph and Network Security by B. A. Forouzan & D. Mukhopadhyay, 11th edition, McGraw Hill Publication

Reference Books :

1. Elementary Number Theory and its applications by Kenneth H. Rosen, 5th edition, Addison Wesley Publication
2. Advanced Engg. Mathematics by C. Ray Wylie & Louis Barrett. TMH International Edition
3. C. L. Liu: Elements of Discrete Mathematics, 2nd edition, TMH
4. MATLAB: An Introduction with Applications by Amos Gilat, Fourth edition, John Wiley and Sons(Wiley)

Other Resources :

1. NPTEL Course: Applied Linear Algebra IIT Madras: Dr Andrew Thangaraj: :-Web link- https://nptel.ac.in/courses/108106171
2. NPTEL Course: Computational Number Theory and Cryptography IIT Guwahati: Dr Pinaki Mishra:-Web link- https://nptel.ac.in/courses/106103015
3. NPTEL Course:NOC: Numerical Methods, IIT Roorkee Prof. Ameeya Kumar Nayak, Prof. Sanjeev Kumar - Web link- https://nptel.ac.in/courses/111107105
4. NPTEL Course: MATLAB Programming for Numerical Computation: Prof. Niket Kaisare, Web link- https://nptel.ac.in/courses/103106118

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment - Theory (20 MARKS)

One MCQ test as per Gate exam pattern/ level: 5 Marks
One Class test: 5 Marks
One Team-pair- Solo: 5 Marks
Regularity and attentiveness: 5 Marks

2. Mid Semester Exam (30 Marks)

Mid semester examination will be based on 40% to 50% of the syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20%-30% weightage, and the syllabus covered from MSE to ESE carrying 70%-80% weightage.

Course Type	Course Code	Course Name	Credits
PCC	CSPCC302	DISCRETE STRUCTURES	03+01*

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Examination (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20 + 25*	30	50	1.5	2	125

* ***Tutorial***

Pre-requisite:

1. BSC101- Engineering Mathematics-I
2. BSC204-Engineering Mathematics-II

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/development of solutions
4. PO6: The Engineer and the world
7. PO7: Ethics

Course Objectives:

1. To inculcate the notion of Set Theory .
2. To familiarize with the concepts of Propositional logic.
3. To familiarize with the concepts of Relations ,Posets and Lattice.
4. To familiarize with the concepts of Functions , Euler and Hamiltonian path/circuits.
5. To acquaint learners with Algebraic Structures and Coding Theory.
6. To familiarize with the concepts of Combinatoric and Recurrence Relations.

Module	Details	Hrs
00	**Course Introduction** Discrete Structures course strengthens analytical and problem-solving skills essential for algorithm design, network analysis, cryptography, and software engineering, thereby fostering professional growth in the field of computer science and engineering.	**01**
01	**Sets:** ***Learning Objective:*** *Students are expected to apply concepts of sets including types of sets, set operations and properties, power sets, and the Principle of Inclusion and Exclusion to solve discrete structure problems.*	**04-06**

	Contents: Introduction to Set Theory, Venn diagrams, Operations on Sets, Power sets, Laws of set theory, Cartesian Product, Partitions of sets, The Principle of Inclusion and Exclusion , Application in Database Systems and data mining	
	Self-Learning Topics: *Advanced applications of Venn diagrams in problem solving*	
	Learning Outcomes: *A learner will be able to* *LO1.1: Apply fundamental concepts of set theory, Venn diagrams, types of sets, and set representations. in problem solving. (P.I.- 1.3.1)* *LO1.2: Perform set operations, apply laws of set theory, and construct power sets and Cartesian products for problem solving. (P.I.- 1.1.1)* *LO1.3: Analyze partitions of sets and apply the Principle of Inclusion and Exclusion to solve counting and logical problems. (P.I.- 2.1.3, P.I. ,6.3.1)* *LO1 .4 Identify set theory concepts and apply it in real-world applications such as database systems and data mining. (P.I.- 2.2.3, P.I. ,6.3.2)*	
2.	**Propositional Logic:** ***Learning Objective:*** *Students are expected to apply the fundamentals of propositional logic, including logical connectives, truth tables, logical equivalence, and inference rules to analyze and solve logical problems in discrete structures.*	**04-06**
	Contents: Propositions and Logical operations, Truth tables, Laws of Logic, Equivalence, Implications, Predicates, Quantifiers, Application in Digital Circuit Design and Artificial Intelligence & Expert Systems	
	Self-Learning Topics: *Normal forms of logical expressions, Applications of propositional logic in formal verification and software testing.*	
	Learning Outcomes:	
	Learning Outcomes: *A learner will be able to* *LO2.1: Apply propositional logic concepts to solve real-time computing problems. (P.I.- 1.1.1 ,P.I. ,6.3.1)* *LO2 .2 Apply predicate logic to model and solve real-world engineering and societal problems effectively. (P.I.- 1.3.1, P.I. ,6.3.2)*	
3.	**Relations, Posets and Lattice** ***Learning Objective:*** *Apply concepts of relations, partially ordered sets (posets), and lattices to analyze ordered structures and solve discrete mathematics problems.*	**11-13**
	Contents: Introduction to Relation and its properties, Paths and Digraphs, types of binary relations, Operations on relations, Equivalence relations, Closures, Warshall's algorithm, Application of relation in Social Media Networks, Recommendation Systems, Partial Order Relations, Poset, Hasse Diagram, Chain and Anti chains,	

	Application Poset in Task Scheduling and Dependency Management Database Query Optimization Lattice, Types of Lattices, Sub lattice, Application of lattice in real life e.g. Boolean Algebra and Logic Circuits , Information Flow and Security Models	
	Self-Learning Topics: *Isomorphism in Lattice.*	
	Learning Outcomes: *A learner will be able to* *LO3.1 Apply properties and operations of relations, equivalence relations, closures, and Warshall's algorithm to analyze paths, digraphs, and real-world network problems.(P.I.-1.4.1)* *LO3.2 Apply concepts of lattices, types of lattices, and sublattices to solve problems in Boolean algebra, logic circuits, and information flow systems. .(P.I.- 1.3.1)* *LO3.3:Design efficient and ethical solutions using concepts of posets and lattices for task scheduling, query optimization, and secure information flow. (P.I.3.1.1, P.I. ,7.1.1)* *LO3.4:Develop lattice-based models for logical circuit design and data dependency management that address societal and ethical aspects of engineering solutions.(P.I.3.4.3, P.I. ,7.2.2)*	
4.	**Functions, Euler and Hamiltonian paths/Circuit**	**05-07**
	Learning Objective: *Apply concepts of functions, Euler graphs, and Hamiltonian graphs to model relationships and analyze connectivity in graph-based problems.*	
	Contents: Functions,Types of functions - Injective, Surjective and Bijective, Composition of functions, Identity and Inverse function, Euler and Hamiltonian Theorem for path /circuit, Planar Graph , Applications in Computer Science and Programming, Cryptography,Logistics and Transportation	
	Self-Learning Topics: *Real-life application of Euler graphs in network routing . Hamiltonian and Euler graphs assist in path planning, navigation, and decision-making in robotics and intelligent systems.*	
	Learning Outcomes:	
	A learner will be able to *LO4..1: Apply the concepts of functions to represent and solve real-world problems (P.I-1.1.2)* *LO4.2: Apply the theorems to identify Euler/Hamiltonian path/circuit in graph. (P.I.- 1.3.1)* *LO4.3: Apply function types to design secure and ethical models in networking and cryptography. (P.I.-1.4.1, P.I.-3.2.1,P.I. 7.1.1)* *LO4.4: Develop and analyze inverse and composite function-based solutions with social and ethical responsibility. (P.I.-1.3.1, P.I.-3.1.1,P.I. 7.2.2)*	
05.	**Algebraic Structures and Coding Theory**	**09-11**
	Learning Objective/s: *To familiarize students with algebraic structures and enable them to design solutions for real-life problems.*	
	Contents: Algebraic structures with one binary operation - semi groups, monoids and groups, Abelian group, Isomorphism, homomorphism, Cyclic groups, Normal sub group, codes and group codes, Application of algebraic structure in Cryptography and Network Security, error detection and correction, Automata Theory and Formal Languages.	
	Self-Learning Topics: *Co-sets and Lagrange's theorem , Product and quotient of algebraic structures*	
	Learning Outcomes :	

	A learner will be able to *LO5.1: Identify Semi-group, monoid, groups, subgroups and abelian based on the binary operation. (P.I.- 1.1.1)* *LO5.2: Identify whether the given algebraic structures are isomorphic/ homomorphic ((P.I.- 1.4.1)* *LO5.3 Analyze algebraic structures to solve problems in cryptography and network security. (P.I.-2.4.1,6.1.1,)* *LO5.4 Apply algebraic concepts to design secure, reliable, and socially responsible communication systems .(P.I.-2.1.3,6.3.2)*	
06.	**Combinatoric and Recurrence relation**	**04-06**
	Learning Objective/s: *Students are expected to solve recurrence relations and apply advanced concepts of counting to provide the solution to associated problems*	
	Contents: Permutations and combinations, Pigeon-hole principle, Calculating Coefficient of generating function, Recurrence relations,Solving recurrence relation by substitution and Generating functions	
	Self-Learning Topics: *Permutation and combination*	
	Learning Outcomes: *A learner will be able to* *LO6.1: Use the concept of Permutations and combinations to solve counting problems. (P.I.- 1.1.2)* *LO6.2: Solve recurrence relation using substitution and Generating functions .(P.I.- 1.3.1)* *LO6.3: Design efficient algorithms using recurrence relations and combinatorial techniques for real-world problem solving. (P.I.- 3.1.1 ,6.1.1)* *LO6.4: Apply combinatorial and algorithmic principles to develop solutions that address societal needs responsibly. (P.I.- 3.4.3,6.3.2)*	
	Course Conclusion In conclusion, proficiency in discrete structures is invaluable for problem-solving in computer engineering, facilitating career growth and impactful contributions across diverse domains.	**01**
	Total	**45**

Performance Indicators:

P.I. No.	P.I. Statement
1.1.1	Apply the knowledge of discrete structures, linear algebra, statistics and numerical techniques to solve problems.
1.1.2	Apply the concepts of probability, statistics and queuing theory in modelling of computer-based system, data and network protocols.
1.3.1	Apply engineering fundamentals.
1.4.1	Apply theory and principles of computer science and engineering to solve an engineering problem
2.1.3	Identify mathematical algorithmic knowledge that applies to a given problem
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions
2.4.1	Applies engineering mathematics to implement the solution.
3.1.1	Able to define a precise problem statement with objectives and scope.
3.2.1	Ability to explore design alternative
3.4.3	Able to verify the functionalities and validate the design.
6.1.1	Identify and describe various engineering roles; particularly as pertains to protection of the public and public interest at global, regional and local level
6.3.1	Identify risks/impacts in the life-cycle of an engineering product or activity
6.3.2	Understand the relationship between the technical, socio economic and environmental dimensions of sustainability
7.1.1	Identify situations of unethical professional conduct and propose ethical alternatives
7.2.2	Examine and apply moral & ethical principles to known case studies

Course Outcomes: A learner will be able to -

1. Solve the basic problems of Set theory . (*LO1.1, LO1.2, LO1.3, LO1.4)*
2. Apply the fundamentals of propositional logic to solve logical problems in discrete structures. *(LO2.1, LO2.2)*
3. Apply basic principles of, Relation, Posets and Lattice to find the solution to computer science & engineering problem. (*LO3.1, LO3.2, LO3.3, LO3.4)*
4. Apply concepts of functions, Euler and Hamiltonian paths/circuit to model relationships in graph-based problems. *(LO4.1, LO4.2, LO4.3, LO4.4)*
5. Design solutions to real-world problems in various domains of Computer Science Engineering using the principles of Algebraic Structures and Coding Theory. (*LO5.1, LO5.2, LO5.3, LO5.4)*
6. Solve problems involving Combinatoric and recurrence relations . (*LO56.1, LO6.2, LO6.3, LO6.4)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC302.1	3	3	-	-	-	3	-	-	-	-	-
CSPCC302.2	3	-	-	-	-	3	-	-	-	-	-
CSPCC302.3	3	-	3	-	-	-	3	-	-	-	-
CSPCC302.4	3	-	3	-	-	-	3	-	-	-	-
CSPCC302.5	3	3	-	-	-	3	-	-	-	-	-
CSPCC302.6	3	-	3	-	-	3	-	-	-	-	-
Average	**3**	**3**	**3**	**-**	**-**	**3**	**3**	**-**	**-**	**-**	**-**

Text Books :

1. Discrete Mathematical Structures, Bernad Kolman, Robert Busby, Sharon Cutler Ross, Nadeem-ur-Rehman, Sixth edition , 2015 Pearson Education.
2. Discrete Mathematics and applications, K. H. Rosen, fifth edition 2003, Tata McGraw Hill Publishing Company.
3. Elements of Discrete Mathematics, C. L. Liu, second edition 1985, Reprinted 2000, McGraw-Hill Book Company.

Reference Books :

1. Discrete Mathematics for Computer Scientists and Mathematicians, J. L. Mott, A. Kandel, T. P. Baker, Second Edition 1986, Prentice Hall of India.
2. Discrete Mathematical Structures with Applications to Computer Science, J. P. Trembley, R. Manohar ,2002,Tata McGraw Hill Publishing Company.
3. Graph Theory with applications to engineering and computer science, Narsing Deo, 1st publication 2016, PHI Publications.
4. Discrete Mathematics, P. K. Bisht, H. S. Dhami, 1st publication 2015, Oxford press.
5. Discrete Mathematical Structures, Y N Singh, 2010,Wiley-India.

Other Resources :

1. NPTEL Course: Discrete Mathematics By Prof. Sudarshan Iyengar, Department of Computer Science Engineering, IIT Ropar :-Web link- https://onlinecourses.nptel.ac.in/noc19_cs49/
2. NPTEL Course: Discrete Mathematics By Prof. Sourav Chakraborty,Department of Computer Science Engineering, IIT Madras :-Web link- https:://nptel.ac.in/courses/111/106/111106086/
3. NPTEL Course: Discrete Mathematics By Prof. Ashish Choudhury, Department of Computer Engineering, IIIT Bangalore:- Web link- https://nptel.ac.in/courses/107/106/107106081/

A.IN-SEMESTER ASSESSMENT (75 MARKS)

1. Continuous Assessment - Theory-(20 Marks)

Suggested breakup of distribution

MCQ test strictly as per GATE exam pattern / level): 05 Marks

Class test: 05 Marks

Open book test/ Open notes test: 05 Marks

Regularity and active participation: 05 Marks

2. Continuous Assessment - Tutorial- (25 Marks)

Suggested breakup of distribution

Numerical Assignments: 10 Marks

Class test based on above numerical assignment: 10 Marks

Regularity and active participation: 05 Marks

3. Mid Semester Exam (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20%-30% weightage, and the syllabus covered from MSE to ESE carrying 70%-80% weightage.

Course Type	Course Code	Course Name	Credits
PCC	CSPCC303	DATA STRUCTURES	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Exam (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite:

1. ESL103: Programming Laboratory -I(C)

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of Solutions
4. PO4: Conduct Investigations of Complex Problems
5. PO5: Engineering Tool Usage
6. PO11: Life-Long Learning

Course Objectives:

1. Facilitate the way students, comprehend computer automation in addressing engineering difficulties through the use of data structures.
2. Intended to guide learners through developing complex data structures through the application of fundamental programming concepts.
3. Aid learners to learn the basics of basic data structures including linked lists, stacks, and queues.
4. Assist learners in acquiring a strong comprehension of the fundamental concepts of different types of trees and graphs.
5. Asist learners with applying and using the concepts of searching, hashing, and sorting.
6. Assist learners in applying linked lists, stacks, queues, trees, and graphs to solve complicated issues.

Module	Detailed Contents	Hrs
00.	**Course Introduction** This course deals with basics of data types and variables, revision of concept of array, structures and pointers. Data is simplified by data structures. Data structures organize information for machines and humans to understand, most importantly. Understanding DSA helps to solve real-world problems quickly. Data structures and algorithms support logical problem-solving.	**01**
01.	**Introduction of Data Structures and Algorithms Analysis** ***Learning Objective:*** *Learner is expected to recall and interpret data structures and algorithms to solve real time-complex issues and maximize results.*	**07-09**
	Contents: Introduction of Data Structures: Concept of computation, algorithms, elementary data types, abstract data types and data structures. RAM model of computation. Data Management concepts, Data types – primitive and non-primitive, Types of Data Structures- Linear & Non-Linear Data Structure. Introduction of Algorithm Analysis: Algorithm, pseudo code for expressing algorithms, performance analysis-space complexity, time complexity, asymptotic notation- big (O) notation, omega notation, theta notation and little (o) notation, recurrences, probabilistic analysis, disjoint set operations, union and find algorithms. applications-analysis of binary search.	
	Self-Learning Topics: *Asymptotic Notations, advanced data structures.*	
	Learning Outcomes: *A learner will be able to* *LO 1.1: Apply the concepts of queuing theory in modelling of computer-based system, data and processing. (P.I.-1.1.2)* *LO 1.2: Use concepts of asymptotic notations and apply algorithm to analyze computational performance of given problem statements. (P.I.-1.4.1)* *LO 1.3 Identify appropriate data structures for solving specific computational problems. (P.I.-2.1.2)* *LO 1.4: Identifies processes/modules/algorithms of a computer-based system and parameters to solve a problem by making use of data structures. (P.I.-2.2.2)* *LO 1.5: Design precise problem statement and discuss the required data structures to implement it. (P.I. 3.1.1)* *LO 1.6: Ability to explore design alternatives for various data structures with respect to their mechanism by discussing their limitations. (P.I.-3.2.1)*	
02.	**Linear Data Structure-Stack, Queue, Linked List** ***Learning Objective:*** *Learner is expected to know linear data structures and interpret all linear data and functions.*	**09-11**
	Contents: Array: Representation of arrays, Applications of arrays, sparse matrix and its representation,	

	Stack: Stack-Definitions & Concepts, Operations on Stacks, Applications of Stacks, Polish Expression, Reverse Polish Expression and Their Compilation, Recursion, Tower of Hanoi, Queue: Representation Of Queue, Operations on Queue, Circular Queue, Priority Queue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue. Linked List: Singly Linked List, Doubly Linked list, Circular linked list, linked implementation of Stack, Linked implementation of Queue, Applications of lists.	
	Self-Learning Topics: *Linked List Implementation of Stack, Linked List implementation of Queue, Circular Queue, Double Ended Queue, Priority Queue*	
	Learning Outcomes: *A learner will be able to* *LO 2.1: Apply and use engineering fundamentals to know the linear data management and uses. (P.I.-1.3.1)* *LO 2.2: Apply principle of stack data structure to evaluate real -world problem. (P.I.-1.4.1)* *LO 2.3: Identify existing methods to solve the problem using standard data structures mechanism. (P.I.-2.2.3)* *LO 2.4: Identify the limitations of the solutions and causes w.r.t mechanism of linear data structures. (P.I.-2.4.3)* *LO 2.5: Able to implement and integrate the modules in several applications by exploring and using data structure mechanism. (P.I. 3.2.1, P.I.-3.4.2)* *LO 2.6: Design and develop appropriate methodologies based on the principles of various linear data structures. (P.I.-4.2.1)* *LO 2.7: Use appropriate procedures, techniques to collect and analyze data by applying idenfied data structure's mechanism. (P.I.-4.3.1)*	
03.	**Nonlinear Data Structure-Tree, Graph** ***Learning Objective:*** *Learner is expected to know and interpret non-linear data structures, utilize them, and analyse its many functions.*	**10-12**
	Contents: Tree-Definitions and Concepts, Representation of binary tree, Binary tree traversal (In-order, Post-order, Preorder), Threaded binary tree, Binary search trees, Conversion of General Trees to Binary Trees, AVL trees, Height Balanced, Weight Balance. Insertion and deletion cases of tree. Graph-Types of Graphs, Matrix Representation of Graphs, Elementary Graph operations, (Breadth First Search, Depth First Search, Spanning Trees, Shortest path, MST)	
	Self-Learning Topics: *Implementation of AVL and B+ Tree.*	
	Learning Outcomes: *A learner will be able to* *LO 3.1: Apply the terminology and properties of trees and graphs on system functionalities (e.g., nodes, edges, degree, path, height, etc.). (P.I.-1.4.1)* *LO 3.2: Identifies benefits of non-linear data structures to solve a problem. (P.I. 1.1.2, P.I.-2.1.2)*	

	LO 3.3: Identify the limitations of the solutions and causes w.r.t mechanism of non- linear data structures. (P.I.-2.4.3) *LO 3.4 Ability to explore design alternatives for tree/graph with respect to their mechanism. (P.I.-3.2.1)* *LO 3.5: Identify suitable constraint based on mechanism of tree to design the structures and evaluate for alternate design solutions. (P.I.-3.2.3)* *LO 3.6: Design and develop appropriate procedures/methodologies using appropriate data structure. (P.I.-4.2.1)* *LO 3.7: Synthesize information and knowledge about the problem from the given sequence of data element to reach appropriate conclusions (P.I.-4.3.4, 11.2.1)* *LO 3.8: Recognize the need of data structure and identifies the importance of its current regarding new developments of emerging technologies (P.I. 4.1.2, P.I.-11.2.2)*	
04.	**Sorting and searching** ***Learning Objective:*** *Learner is expected to recall searching and sorting algorithm. Also expected to apply it to order the data with list and arrays.* **Contents:** Introduction of Sorting methods, Concept, Insertion Sort, Quick Sort, Merge Sort, Selection Sort, Heap Sort, Sorting on Several Keys, List and Table Sort. Introduction of Searching concept, Linear Search, Binary Search, Complexity Comparison. ***Self-Learning Topics:*** *Implementation of different sorting techniques and searching.* ***Learning Outcomes:*** *A learner will be able to* *LO 4.1: Apply the working mechanism of basic sorting algorithms such as Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort, and Heap Sort. (P.I.-1.4.1)* *LO 4.2: Evaluate problem statements and identifies objectives of data organization. (P.I.-2.1.1)* *LO 4.3: Identify the limitations of the solution and causes w.r. t. the applied method. (P.I.-2.4.3)* *LO 4.4 Ability to explore design alternatives for sorting/searching with respect to every methods ability to work on given dataset. (P.I.-3.2.1)* *LO 4.5 Able to verify the mechanism of various sorting functions and validate the method on given set of values. (P.I.-3.4.3)* *LO 4.6: Able to choose appropriate procedure/algorithm, data and analyze the output. (P.I. 4.1.2)* *LO 4.7 Represent data (in tabular and/or graphical forms) using various data analysis methods and find the conclusions. (P.I.-4.3.3, P.I. 11.2.1)* *LO 4.8 Recognize the need of data retrieval and organization for new developments of emerging technologies by applying engineering fundamentals. (P.I. 1.3.1, P.I.-11.2.2)*	**03-05**
05.	**Hashing and File Structures** *Learning Objective:* *Learner is expected to recall and summarize understand hashing and file structures to simplify the process of finding or using the original string.*	**05-07**

	Contents: Hashing: The symbol table, Hashing Functions, Collision-Resolution Techniques, File Structure: Concepts of fields, records and files, Sequential, Indexed and Relative/Random File. Organization, indexing structure for index files, hashing for direct files, multi-key file organization and access methods.	
	Self-Learning Topics: *Implementation of applications of hashing methods and collision techniques.*	
	Learning Outcomes: *A learner will be able to* *LO 5.1: Apply hash functions and their properties such as uniformity and minimized collisions. (P.I.-1.4.1)* *LO 5.2: Identify and apply different collision resolution techniques like chaining, open addressing, linear probing, quadratic probing, and double hashing. (P.I.-2.1.2)* *LO 5.3 Identifies functionalities and computing resources for file and directory structures. (P.I.-2.2.2)* *LO 5.4 Able to explore design solution and compare and contrast alternative solution/methods by selecting proper data structure mechanism. (P.I.-3.2.1)* *LO 5.5: Able to design solutions suited to meet functional requirements using hash tables. (P.I.-3.2.2)* *LO 5.6: Use appropriate techniques to analyze data in case of hashing (P.I.-4.3.1, P.I. 11.2.1)* *LO 5.7: Modify/ Formulate hash function and techniques to get appropriate solutions for problem by critically analyzing data. (P.I. 1.3.1, P.I.-4.3.2, P.I. 11.2.2)*	
06.	**Data Structures in Real-World Systems** ***Learning Objective/s:*** *This chapter focuses on how to develop the ability to select and implement appropriate data structures for solving practical computational problems and to gain insight into how abstract data types are mapped to real-world data models and system-level functionalities.*	**03-05**
	Contents: Data structures in: Databases (B+ Trees, Hash Indexing), Compilers (Syntax trees, symbol tables), Networks (Routing tables, adjacency structures), Machine learning (Decision trees, KD-trees) Case studies: examples of complex computational challenges and data structure solutions	
	Self-Learning Topics: *Implementation of applications for Stack, Queues, Linked List, Trees and Graph.*	
	Learning Outcomes: *A learner will be able to* *LO 6.1: Identifies data structure algorithms and parameters to solve any computational problem with high level of efficiency by applying data structure mechanism. (P.I. 1.4.1, P.I.-2.1.2)* *LO 6.2: Identify the limitations of the solution and causes w.r. t. the applied method. (P.I.-2.4.3)* *LO 6.3: Ability to explore design alternatives of selected and applied method. (P.I.-3.2.1)* *LO 6.4: Ability to verify the mechanism of various data structures functions and validate the method on given problem statement. (P.I.-3.4.3)* *LO 6.5: Ability to choose appropriate procedure/algorithm, data set and test cases by applying engineering fundamentals. (P.I. 1.3.1, P.I.-4.1.2)*	

	LO 6.6: Design and develop appropriate procedures/methodologies based on the mechanism and usage of data structures. (P.I.-4.2.1, P.I. 11.2.1) *LO 6.7: Modify tools and techniques to solve engineering problems to recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in the field. (P.I.-5.1.2, P.I. 11.2.2)*	
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

P.I. No.	P.I. Statement
1.1.2	Apply the concepts of probability, statistics and queuing theory in modelling of computer based system, data and network protocols.
1.3.1	Apply engineering fundamentals.
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem.
2.1.1	Evaluate problem statements and identifies objectives.
2.1.2	Identifies processes/modules/algorithms of a computer.
2.2.2	Identifies functionalities and computing resources.
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions.
2.2.4	Compare and contrast alternative solution/methods to select the best methods
2.4.3	Identify the limitations of the solution and sources/causes.
3.2.1	Ability to explore design alternatives.
3.2.2	Ability to produce a variety of potential design solutions suited to meet functional requirements.
3.2.3	Identify suitable non-functional requirements for evaluation of alternate design solutions.
3.4.2	Ability to implement and integrate the modules.
3.4.3	Ability to verify the functionalities and validate the design.
4.1.2	Ability to choose appropriate procedure/algorithm, data set and test cases.
4.2.1	Design and develop appropriate procedures/methodologies based on the study objectives.
4.3.1	Use appropriate procedures, tools and techniques to collect and analyze data.
4.3.3	Represent data (in tabular and/or graphical forms) so as to facilitate analysis and explanation of the data, and drawing of conclusions.
4.3.4	Synthesize information and knowledge about the problem from the raw data to reach appropriate conclusions.
5.2.1	Identify the strengths and limitations of tools for (i) acquiring information, (ii) modeling and simulating, (iii) monitoring system performance, and (iv) creating engineering designs.
5.3.1	Discuss limitations and validate tools, techniques and resources.
11.1.2	Analyze different forms of financial statements to evaluate the financial status of an engineering project.
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field.

Course Outcomes:

Learner will be able to

1. Interpret the concepts of algorithm analysis, time complexity, space complexity, and abstract data types. *(LO 1.1,LO 1.2, LO 1.3,LO 1.4,LO 1.5, LO 1.6)*
2. Implement linear data structures such as arrays, linked lists, stacks, and queues. *(LO 2.1,LO 2.2, LO 2.3,LO 2.4, LO 2.5. LO 2.6,LO 2.7)*
3. Design and analyze non-linear data structures such as trees and graphs. *(LO 3.1,LO 3.2, LO 3.3,LO 3.4, LO 3.5, LO 3.6, LO 3.7, LO 3.8)*
4. Implement sorting, searching, and hashing algorithms to solve computational problems efficiently. (*LO 4.1,LO 4.2, LO 4.3,LO 4.8, LO 5.1, LO 5.2, LO 5.3, LO 5.4, LO5.7)*
5. Develop efficient solutions using suitable data structures for real-world or project-based problems. (*LO 6.1,LO 6.2, LO 6.3,LO 6.4, LO 6.5,LO 6.6, LO 6,7)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC303.1	3	3	3	-	-	-	-	-	-	-	-
CSPCC303.2	3	3	3	3	-	-	-	-	-	-	-
CSPCC303.3	3	3	3	3	-	-	-	-	-	-	3
CSPCC303.4	3	3	3	3	-	-	-	-	-	-	3
CsPCC303.5	3	3	3	3	3	-	-	-	-	-	3
Average	3	3	3	3	3	-	-	-	-	-	3

Text Books:

1. Data Structures through C in Depth, S. K Srivastava, Deepali Srivastava, 5th Edition, 2011, BPB Publications.
2. Data Structures Using C by Aaron M Tenenbaum, 1st Edition ,2018, Pearson India.
3. Data Structures using C, Reema Thareja, 2nd Edition, 2011, Oxford.

Reference Books:

1. Fundamentals of Data Structures, Ellis Horowitz, Sartaj Sahni, 5th Edition, 2010, Galgotia Publications.
2. An introduction to data structures with applications, Jean Paul Tremblay, Paul G. Sorenson; 3rd Edition, 1984, Tata McGraw-Hill.
3. Data Structures using C and C++, Rajesh K. Shukla, 2nd Edition, 2009, Wiley India.

Other Resources:

Digital material:

1. Web Link-Complete DS Data Structure in one shot | Semester Exam | Hindi (youtube.com)
2. Web Link-https://topperworld.in/dsa-handwritten-notes/

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (20 Marks)

Suggested breakup of distribution

a) Numerical Assignment/s (Minimum 20 problems) : 05 Marks

b) Class Test based on above numerical assignment: 05 Marks

c) One Flip classroom activity: 05 Marks

d) Regularity and active participation :05 Marks

2. Mid Semester Examination (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20% to 30% weightage, and the syllabus covered from MSE to ESE carrying 70% to 80% weightage.

Course Type	Course Code	Course Name	Credits
PCC	CSPCC304	DATABASE MANAGEMENT SYSTEM	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Examination (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite:

1. ESL103: Programming Laboratory-I (C)

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/development of solutions
4. PO8: Individual and Collaborative team work
5 PO9: Communication
6 PO11: Life-long learning

Course Objectives:

1. To impart database concepts, including database structures, levels of abstraction, ER modeling, EER modeling, and relational schema design for real-world applications
2. To introduce SQL commands for data definition (DDL), data manipulation (DML), integrity constraints, security management (DCL), indexing, and query optimization for effective database management.
3. To impart Knowledge **of** functional dependencies, schema decomposition, and normalization techniques (1NF, 2NF, 3NF, BCNF) to enhance database efficiency and integrity.
4. To acquaint knowledge of database transaction management, ACID properties, concurrency control mechanisms, deadlock handling, and recovery techniques to ensure data consistency and security.
5. Impart knowledge to students on working collaboratively in teams, communicating database design decisions, and applying security, indexing, and optimization techniques to real-world database solutions

Module	Details	Hrs.
00.	**Course Introduction**	**01**
01.	**Introduction to DBMS** ***Learning Objective:***	**04-05**

	To impart Learner to recall the concept of database system architecture and interpret the process of data management	
	Contents: Introduction, Difference between data and information, Characteristics of Databases, File system v/s Database system, Data abstraction and Physical Data Independence, DBMS system architecture, Database Administrator (DBA), Role of DBA, significance of Relational Database	
	Self-Learning Topics: *Identify the types of Databases.*	
	Learning Outcomes: *A learner will be able to* *LO 1.1: Apply engineering fundamentals to summarize the key characteristics of databases, compare them with file systems, and evaluate the advantages of databases in terms of data integrity, security, scalability, and efficiency in data management. (PI- 1.3.1)* *LO 1.2: Evaluate problem statements and identify objectives to classify different levels of abstraction in database systems, applying a structured approach to examine their roles in data organization, access, and management.. (PI- 2.1.1)* *LO 1.3: Apply theory and principles of computer science engineering to solve problems related to drawing and interpreting the structure of a database system, analyzing its components and their interactions to understand overall database functionality and management.. (PI-1.4.1)* *LO 1.4: Identify the key processes, modules, and algorithms within a database system to apply physical data independence effectively, ensuring that changes in physical storage do not impact logical data organization or application programs. (PI-2.1.2)*	
02.	**E-R model** ***Learning Objective:*** *To introduce learner recall and demonstrate the concept of ER model. Also expected to apply the concept of ER model for conceptual design of database*	**05-07**
	Contents: Conceptual Modeling of a database, The Entity-Relationship (ER) Model, Entity Type, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Weak entity Types. Extended Entity-Relationship (EER) Model: Generalization, Specialization and Aggregation. Design an ER model for real time case study. (e.g. Hospital management system)	
	Self-Learning Topics: *Design an ER model for any real time case study*	
	Learning Outcomes: *A learner will be able to* *LO 2.1: Evaluate problem statements and identify objectives to recognize and categorize entity sets and relationship sets, applying a structured approach to represent relationships effectively in database design.. (PI-2.1.1)* *LO 2.2: Identify the relevant processes, modules, and parameters within a database system to implement constraints and mapping cardinalities effectively, ensuring data integrity and consistency by applying structured approaches to enforce database constraints.. (PI- 2.1.2)*	

	LO 2.3: Define a precise problem statement with clear objectives and scope to develop Entity–Relationship (ER) diagrams for real-time problems, applying structured engineering approaches to represent entities, attributes, relationships, and constraints accurately in database design (PI- 3.1.1) *LO 2.4: Identify and document system requirements from stakeholders to design Enhanced Entity–Relationship (EER) diagrams for complex real-time problems, applying structured approaches to represent specialization, generalization, and inheritance in database systems. (PI- 3.1.2)* *LO 2.5: Apply engineering fundamentals to develop conceptual database models using the Entity–Relationship (ER) approach by identifying entity types, attributes, keys, relationship sets, and weak entities to ensure structured data representation. (PI- 1.3.1)* *LO 2.6: Apply theory and principles of computer science engineering to solve engineering problems by creating ER and EER models that incorporate generalization, specialization, and aggregation, and designing comprehensive database models for real-time case studies such as a hospital management system.. (PI- 1.4.1)*	
03.	**Introduction to Relational Model** ***Learning Objective:*** *To impart learner recall and demonstrate the process of conversion of ER into relational model. Also expected apply the process to create a database.*	**05-07**
	Contents: Relational Model Constraints and Relational Database Schemas, Concept of Keys: Primary Kay, Secondary key, Foreign Key, Introduction to Relational Algebra, Relational Algebra expressions for Unary Relational Operations, • Set Theory operations, • Binary Relational operation Relational Algebra Queries, Conversion of ER model in to relational schema by using rules. (e.g. Hospital management system)	
	Self-Learning Topics: *Convert the ER model designed to relational schema by using rules* for real time problem	
	Learning Outcomes: *A learner will be able to* *LO 3.1: Apply engineering fundamentals to create conceptual database models using the Entity–Relationship (ER) approach by identifying and classifying entity types, entity sets, attributes, keys, and relationship types for structured data representation. (PI-1.3.1)* *LO 3.2: Apply theory and principles of computer science engineering to solve problems related to identifying and interpreting various relationship types, including weak entity types, and ensuring proper representation of data relationships in database models.. (PI- 1.4.1)* *LO 3.3:Evaluate problem statements and identify objectives to analyze and categorize entity sets and relationship sets based on their characteristics and functionalities, applying a structured approach to represent relationships effectively in conceptual database design. (PI- 2.1.1)* *LO 3.4: Identify the relevant processes, modules, algorithms, and parameters within a database system to apply integrity constraints and mapping cardinalities effectively, ensuring data consistency and accurate representation of business rules in ER models. (PI- 2.1.2)*	

	LO 3.5: Develop ER models to solve real-world problems, such as a hospital management system. Apply structured techniques to accurately model data, relationships, and constraints. Collaborate effectively in teams, communicate database design decisions clearly, and utilize modern tools to enhance problem-solving in database modeling. (PI- 3.1.1, 8.1.1, 9.1.1, 11.2.1) *LO3.6: Enhance ER models by incorporating Extended Entity-Relationship (EER) concepts such as generalization, specialization, and aggregation. Apply advanced database modeling techniques to design comprehensive real-time systems. Collaborate effectively in teams, communicate design decisions clearly, and stay updated with evolving database technologies to ensure efficient system implementation. (PI- 3.1.2, 8.1.2, 9.1.3, 11.2.2)*	
04.	**Overview of SQL** ***Learning Objectives:*** *To introduce learner recall and illustrate SQL and also expected to build Queries* **Contents:** Overview of SQL, Data Definition Commands, set operations, aggregate function, null values Data Definition language(DDL), Data Manipulation commands(DML), Data Control commands(DCL), Complex Retrieval Queries using Group by, Recursive Queries, nested Queries, Integrity constraints in SQL, Security and authorization: Grant & Revoke in SQL. Indexing: Basic Concepts, Ordered Indices, Index Definition in SQL Physical design of database for the relational model designed (e.g. Hospital management system),NOSQL ***Learning Outcomes:*** *A learner will be able to* *LO 4.1: Apply engineering fundamentals to use SQL for defining and managing data by applying Data Definition Language (DDL) and Data Manipulation Language (DML) commands to create, modify, and manipulate relational databases efficiently.(PI- 1.3.1)* *LO 4.2: Apply theory and principles of computer science engineering to solve problems involving SQL set operations, aggregate functions, and the handling of null values, ensuring data integrity and meaningful query results. (PI 1.4.1)* *LO 4.3: Evaluate problem statements and identify objectives to develop advanced SQL queries, including GROUP BY, recursive queries, and nested queries, for efficient retrieval and analysis of relational data. (PI 2.1.1)* *LO 4.4Identify the relevant processes, modules, algorithms, and parameters within a database system to implement and enforce integrity constraints in SQL, ensuring data consistency and accuracy in relational databases. (PI 2.1.2)* *LO 4.5: Define a clear problem statement to manage SQL database security using GRANT and REVOKE, collaborate by following teamwork norms, interpret technical and non-technical security information, and stay updated with technological advances to ensure secure multi-user access. (PI: 3.1.1, 8.2.1, 9.1.1, 11.2.1)* *LO 4.6: Identify and document stakeholder requirements to apply indexing techniques for database performance, present integrated team results clearly, create a logical flow in explanations, and stay current with advances in database optimization. (PI: 3.1.2, 8.3.1, 9.1.3, 11.2.2)*	**11-12**
05.	**Normalization** ***Learning Objective/s:*** *To introduce learner define and interpret the concept of normalization and also expected to convert the given relation in different normalization forms to remove redundancy.*	**06-08**

	Contents: Design guidelines for relational Schema, Functional Dependencies, Database tables and normalization, the need for normalization, the normalization process, Improving the design, Definition of Normal Forms- 1NF, 2NF, 3NF & The Boyce-Codd Normal Form (BCNF).	
	Self-Learning Topics: *Consider any real time application and apply normalization up to 3NF/BCNF to remove anomalies.*	
	Learning Outcomes :	
	A learner will be able to *LO 5.1: Apply engineering fundamentals to implement best practices for designing relational schemas that ensure data consistency, integrity, and efficiency in database management. (PI-1.3.1)* *LO 5.2: Apply theory and principles of computer science engineering to solve problems involving the identification and evaluation of functional dependencies in order to establish attribute relationships and strengthen database structure. (PI: 1.4.1)* *LO 5.3: Evaluate problem statements and identify objectives to understand the need for normalization and apply normalization techniques to improve data organization and eliminate redundancy in relational databases. (PI: 2.1.1)* *LO 5.4: Identify the processes, modules, algorithms, and parameters within a database system to apply structured methodologies for decomposing relational tables and improving database design through normalization. (PI: 2.1.2)* *LO 5.5: Define a clear problem statement to apply 1NF, 2NF, and 3NF for refining relational schemas; work effectively by following teamwork norms; interpret technical and non-technical information on normalization; and stay updated with advances in database systems to ensure data integrity and efficiency. (PI: 3.1.1, 8.2.1, 9.1.1, 11.2.1)* *LO 5.6: Identify and document system requirements from stakeholders to apply BCNF and advanced normalization techniques for eliminating anomalies and improving database structure; present results as a team with well-integrated contributions; create a logical flow when communicating database optimizations; and recognize the importance of staying current with advances in database design. (PI: 3.1.2, 8.3.1, 9.1.3, 11.2.2)*	
06.	**Transaction**	**06-08**
	Learning Objective/s: *To introduce learner define and illustrate transaction concept, Concurrency Control and Recovery. Also expected to apply the concept to ensure database consistency.*	
	Contents: Transaction concept, State Diagram, ACID Properties, Transaction Control Commands(TCL), Concurrent Executions, Serializability – Conflict and View, Concurrency Control: Lock-based-protocols, Deadlock handling Timestamp-based protocols, Recovery System: Shadow Paging Recovery Concepts, Log based recovery.	
	Self-Learning Topics: *Study the various deadlock situation which may occur for a database designed in module 5*	
	Learning Outcomes:	

	A learner will be able to *LO 6.1: Interpret the state diagram of transactions and apply Transaction Control Commands (TCL) such as COMMIT, ROLLBACK, and SAVEPOINT to manage database transactions effectively. (PI-1.4.1)* *LO 6.2: Interpret the state diagram of transactions and apply Transaction Control Commands (TCL) such as COMMIT, ROLLBACK, and SAVEPOINT to manage database transactions effectively. (PI: 1.3.1)* *LO 6.3: **Evaluate the problem statement** of concurrent transaction execution in multi-user database environments, **identify the core objectives** of maintaining data consistency and isolation, and select the appropriate criteria (PI- 2.1.1)* *LO 6.4: Identify the processes and modules (concurrent transactions) of a multi-user database system, and identify the algorithms (concurrency control protocols, such as **lock-based protocols** and **timestamp-based protocols**) and **parameters** (deadlock detection/prevention strategies) required to **solve the problem** of managing transaction conflicts for smooth and efficient system execution. (PI- 2.1.2)*	
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

P.I. No.	**P.I. Statement**
1.3.1	Apply fundamental engineering concepts to solve engineering problems.
1.4.1	Apply electrical engineering concepts to solve engineering problems.
2.1.1	Evaluate problem statements and identifies objectives
2.1.2	Identifies processes/modules/algorithms of a *computer based system and parameters to solve a problem*
3.1.1	Able to define a precise problem statement with objectives and scope.
3.1.2	Able to identify and document system requirements from stake holders
8.1.1	Recognize a variety of working and learning preferences; appreciate the value of diversity on a team
8.1.2	Implement the norms of practice (e.g. rules, roles, charters, agendas, etc.) of effective team work, to accomplish a goal
8.2.1	Demonstrate effective communication, problem solving, conflict resolution and leadership skills
8.3.1	Present results as a team, with smooth integration of contributions from all individual efforts
9.1.1	Read, understand and interpret technical and nontechnical information
9.1.3	Create flow in a document or presentation - a logical progression of ideas so that the main point is clear
11.2.1	Identify historic points of technological advance in engineering that required practitioners to seek education in order to stay current
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field

Course Outcomes: A learner will be able to -

1.Apply engineering principles to create ER and EER models, define entity relationships, and optimize database schemas for real-world applications. *(LO 1.1, 1.2, 1.3, 1.4, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 3.1, 3.2, 3.3, 3.4)*

2.Demonstrate proficiency in SQL queries, integrity constraints, security commands (GRANT, REVOKE), indexing, and query optimization to efficiently manage databases. *(LO 4.1, 4.2, 4.3, 4.4, 4.5, 4.6)*

3.Evaluate functional dependencies and apply normalization techniques (1NF, 2NF, 3NF, BCNF) to refine relational schemas and eliminate redundancy. *(LO 5.1, 5.2, 5.3, 5.4, 5.5, 5.6)*

4.Demonstrate TCL commands, serializability techniques, concurrency control protocols, deadlock handling, and recovery mechanisms to manage transactions securely. *(LO 6.1, 6.2, 6.3, 6.4)*

5. Apply security best practices, optimize query execution, improve database performance using indexing and normalization and work collaboratively on database projects,. *(LO 3.5, 3.6, 4.5, 4.6, 5.5, 5.6, 6.3, 6.4)*

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC304.1	3	3	3	-	-	-	-	-	-	-	-
CSPCC304 .2	3	3	3	-	-	-	-	3	3	-	3
CSPCC304.3	3	3	3	-	-	-	-	3	3	-	3
CSPCC304 .4	3	3	-	-	-	-	-	-	-	-	-
CSPCC304.5	-	3	3	-	-	-	-	3	3	-	3
Average	**3**	**3**	**3**	-	-	-	-	**3**	**3**	**-**	**3**

Text Books :

Database System Concepts, Korth, Slberchatz, Sudarshan , Sixth Edition, 2011, McGraw Hill Education

Fundamentals of Database Systems, Elmasri and Navathe, Sixth Edition, 2010, Pearson

Database Management Systems, Raghu Ramkrishnan and Johannes Gehrke, ISE Editions, 1997 ,McGraw-Hill Education

Reference Books :

Database Systems Design, Implementation and Management , Peter Rob and Carlos Coronel, 9th Edition, 2009, Thomson Learning

SQL & PL / SQL for Oracle 11g Black Book, Dr. P. S. Deshpande, ISE Edition, 2011,Dreamtech Press

Database Management Systems , G. K. Gupta , 1st edition,2011,McGraw – Hill

Other Resources :

NPTEL Course on Database Management system by Prof.P.P.Das, Department of Computer Science and Engineering, IIT Khargpur

Web link- https://archive.nptel.ac.in/courses/106/105/106105175/

Coursera course on Introduction to Relational Databases
Web link- https://www.coursera.org/learn/introduction-to-relational-databases#modules

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (20 Marks)

Suggested breakup of distribution

a) One Assignment on live problems/ case studies: 10 Marks

Students should be assigned a real life problem statement (different for each student). Students are expected to research and collect required resources to create a backend for the selected problem. Students should prepare a presentation of 10-15 minutes. This assignment should be graded for 10 marks depending on the parameters as analysis, design, conversion to relational schema and database creation for selected problem statement.

b) Think Pair Share (TPS) activity: 05 Marks

c) Regularity and active participation :05 Marks

1. Mid Semester Exam (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20%-30% weightage, and the syllabus covered from MSE to ESE carrying 70%-80% weightage.

Course Type	Course Code	Course Name	Credits
LBC	CSLBC301	DATA STRUCTURES LABORATORY	01

Examination Scheme		
Continuous Assessment	**Practical /Oral**	**Total**
25	25	50

Pre-requisite:

1. ESL103: Programming Laboratory –I (C)

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of Solutions
4. PO4: Conduct Investigations of Complex Problems
5. PO5: Engineering Tool Usage
6. PO11: Life-Long Learning

Course Objectives:

1. To implement and experiment with fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
2. To develop problem-solving skills through the design and analysis of algorithms using suitable data structures.
3. To use memory management concepts and the efficiency of different data structures in practical scenarios.
4. To enhance programming proficiency by applying data structures to solve real-world problems.
5. To gain hands-on experience in debugging, testing, and evaluating the performance of data structure implementations.

Module	Detailed Contents	Hrs
00.	**Course Introduction** This course is intended to teach the design and analysis of basic data structures and their implementation in an structured-oriented language. The data structures concepts are increasingly becoming the default choice of the IT industry especially industries involved in software development at system level.	**--**
01.	**Implementation of basic data structures** ***Learning Objective/s:*** *Learner is expected to apply unions, pointers, 1D arrays, and structures to solve a given problem.* **Task 1:** Execute following given structure for student data. Use concept of array within structure and array of structure, Demonstrate structure with ten student records in tabular form.	**04**

	struct student { int sno; char sname[20]; int marks[3]; }s[10]; ***Learning Outcomes:*** *A learner will be able to* *LO 1.1: Apply the array, structure concept and implement student record with static memory, Identify and analyze uses of array and able to explore design alternatives by choosing appropriate procedure. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1)* **Task 2:** Execute pointers with structures, array and functions by making use of 2D and 3D pointers for creating references and code for self-referential structure using integration of structure and pointers. ***Learning Outcomes:*** *A learner will be able to* *LO 1.2: Identify and apply best approaches for employing arrays, structures, and pointers, compare and contrast various solutions and methodologies. Analyze the uses of pointers and state the advantages of self-referential structures in dynamic memory allocation.* *(P.I. 1.3.1, P.I.-1.4.1, P.I. 2.1.2, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1)* ***Self-Learning Topics:*** *Pointer implementation of all integrated data types.*	
02.	**Implementation of Static Linear DS** ***Learning Objective:*** *Learner is expected to apply linear data structures and their application in fixed-size scenarios through the use of arrays.* **Task 3:** Create a menu-driven application to construct a static stack with push, pop, peek, and show functions. Write code and measure time complexity. ***Learning Outcomes:*** *A learner will be able to* *LO 2.1: Apply the static stack concept using array allocation to devise a stack architecture and its associated functions and identify computational advantages and disadvantages. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1,P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1)* **Task 4:** Develop a menu-driven application to implement static and circular queues of size 10, incorporating operations for insertion, deletion, retrieval (peek), and display. Write menu driven program using array. ***Learning Outcomes:*** *A learner will be able to* *LO 2.2: Identifies mathematical algorithmic knowledge that applies to design and implement static queue using array, identify limitations of using array in execution of circular queue by choosing appropriate logical statements and functions. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1)* ***Self-Learning Topics:*** *Implement character stack, stack applications.*	**06**
03.	**Implementation of Linear Data Structure using Dynamic Memory Allocation** ***Learning Objective:*** *Learners is expected to comprehend the primary advantage of linked lists compared to arrays and be proficient in using dynamic memory allocation.* **Task 5:** Develop a program to implement the Singly Linked List (SLL**)** data structure and perform various operations on it. The program should allow dynamic creation of nodes, insertion and deletion of elements, and traversal of the linked list. It must include menu-driven functionality to perform the following operations:	**08**

	1. Create a new node and add it to the list. 2. Insert a node at the beginning of the list. 3. Insert a node at the end of the list. 4. Insert a node at a specified position in the list. 5. Delete a node from the beginning of the list. 6. Delete a node from the end of the list. 7. Delete a node from a specified position. 8. Search for an element in the linked list. 9. Count the total number of nodes in the list. 10. Display all elements present in the linked list. ***Learning Outcomes:*** *A learner will be able to* *LO 3.1: Apply theory and principles Singly Linked list to model various functions in application development and identify suitable functionalities to design the LL with a variety of operations and compare it to other types. Comapre and contrast static and dynamic memory allocation by choosing appropriate functions for implementation and explore on design alternatives.* *(P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1)* **Task 6:** Develop a program to implement the Doubly Linked List (SLL**)** data structure and perform various operations on it. The program should allow dynamic creation of nodes, insertion and deletion of elements, and traversal of the linked list. It must include menu-driven functionality to perform various insertion and deletion functions on it. ***Learning Outcomes:*** *A learner will be able to* *LO 3.2: Apply theory and principles Doubly Linked list to model various functions in application development and identify suitable functionalities to design the LL with a variety of operations and compare it to other types. Comapre and contrast static and dynamic memory allocation by choosing appropriate functions for implementation and explore on design alternatives.* *(P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1)* ***Self-Learning Topics:*** *Study and implement header linked list.*	
04	**Implementation of graph and tree** ***Learning Objective:*** *Learner is expected to understand computer file systems and business organizational chart tree structures and implement graph data structures.* **Task 7:** Write a program to implement a **Binary Tree data structure** using dynamic memory allocation. The program should allow the user to perform the fundamental operations on a binary tree, including insert, delete and traversal. ***Learning Outcomes:*** *A Learner will be able to* *LO 4.1: Apply theory and principles tree data structure to model various functions to design and traverse the tree structure for user defined sequence. Comapre time complexity on given functions by choosing appropriate functions for implementation and explore on*	**04**

	design alternatives. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1 P.I. 5.1.2, P.I. 11.2.2, P.I. 11.3.2) **Task 8:** Write a program to represent a graph using an **adjacency matrix** and perform graph traversal using **Breadth-First Search (BFS)** and **Depth-First Search (DFS)**. The program should allow the user to enter the number of vertices and edges, construct the adjacency matrix accordingly, and display the matrix. ***Learning Outcomes:*** *A Learner will be able to* *LO 4.2: Apply theory and principles graph data structure to create graph using 2D array and implement graph traversal with minimum cost. Analyze code performance by comparing functions and construct time efficient procedure for graph strctures. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1 P.I. 5.1.2, P.I. 11.2.2, P.I. 11.3.2)* ***Self-Learning Topics:*** *Implement 2-3, and B and B+ tree mechanism and graph algorithms.*	
05	**Design and Implementation of searching and sorting** ***Learning Objectives:*** *Learner is expected to apply linear data structures and their application in fixed-size scenarios through the use of arrays.* **Task 9:** Develop a program to implement Linear Search and Binary Search techniques on an array of elements. The program should allow the user to input the array elements and the key element to be searched. It must perform the search using both methods and display whether the element is found along with its position. ***Learning Outcomes:*** *A learner will be able to* *LO 5.1: Select and apply suitable searching algorithm on given data set and implement menu driven function-based program. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1 P.I. 5.1.2, P.I. 11.2.2, P.I. 11.2.3)* **Task 10:** Write a program to implement the Quick Sort algorithm using a recursive approach. The program should allow the user to input a list of elements and then apply the Quick Sort technique to arrange the elements in ascending order. ***Learning Outcomes:*** *A learner will be able to* *LO 5.2: Apply quick sorting concepts to a given collection of data and use procedures to create a sorted sequence. Compare the approaches provided to determine the most appropriate strategy. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1 P.I. 5.1.2, P.I. 11.2.2, P.I. 11.2.3)* **Task 11:** Develop a program to implement Insertion Sort and Selection Sort for arranging a list of elements in ascending order. The program should allow the user to input an array of elements and then apply both sorting techniques separately. It must display the array before sorting and after sorting for each method. The implementation should demonstrate a clear understanding of comparison-based sorting, element swapping or shifting, and algorithmic efficiency. The program should also highlight the step-by-step process of arranging elements according to the rules of Insertion Sort and Selection Sort. ***Learning Outcomes:*** *A learner will be able to* *LO 5.3: Apply selection and insertion sorting concepts to a given collection of data and compare both solution and conclude time efficient solution from applied methods.*	**04**

	(P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1 P.I. 5.1.2, P.I. 11.2.2, P.I. 11.2.3)	
	Self-Learning Topics: *Learn to code shell and bucket sort.*	
06	**Implementation of applications of Data structures** ***Learning Objective:*** *Learner is expected to implement complex problems using data structures and algorithms, and analysing data structures inside data management models* **Task 12:** Write a program to implement the infix to postfix expression conversion using the stack data structure. The program should take an infix expression as input and convert it into its equivalent postfix (Reverse Polish Notation) form by applying operator precedence, associativity rules, and proper handling of parentheses. ***Learning Outcomes:*** *A learner will be able to* *LO 6.1. Use stack data structure and apply stack to convert and evaluate various expression types. Compare ability of functions and choose and code with appriate functions by integrating various modules in given problem statement. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1, P.I. 5.1.2, P.I. 11.2.2, P.I. 11.2.3)* **Task 13:** Design and implement dynamic stack and queue using dynamic memory allocation. ***Learning Outcomes:*** *A learner will be able to* *LO 6.2: Apply concept of stack and queue and implement dynamic model for stack and queue. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1, P.I. 5.1.2, P.I. 11.2.2, P.I. 11.3.2)* **Task 14:** Design and implement a Student Database Management System using a self-referential structure in C. The program should support the following operations on student records: 1. Add a new student record 2. Delete an existing student record 3. Display all student records The system must store details of 'N' number of students, where N is entered by the user. Use an appropriate storage design method—static or dynamic memory allocation—to maintain the collection of student records. Based on the chosen design method, justify why this method is more efficient or suitable for storing student data in terms of memory usage, scalability, and flexibility. `struct stud_info` `{ int rno;` `char s_name[25];` `struct stud_info *next;` `}` ***Learning Outcomes:*** *A learner will be able to* *LO 6.3. Apply concept of data structures by selecting appropriate of data structures based on problem requirements, design and develop solution which space and time efficient. Use appropriate tool and utilize methods for modern tools designs. Document all source documents by integrating all procedures of given problem statement. (P.I. 1.3.1, P.I.-1.4.1, P.I. 2.2.4, P.I. 3.2.1, P.I. 3.4.2, P.I. 4.1.2, P.I. 4.2.1, P.I. 5.1.2, P.I. 11.2.2, P.I. 11.3.2)*	**04**
	Self-Learning Topics: *Study advanced data structures for real time problems.*	
	Total	**30**

Performance Indicators:

P.I. No.	P.I. Statement
1.3.1	Apply engineering fundamentals
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem.
2.1.2	Identifies processes/modules/algorithms of a computer.
2.1.3	Identifies mathematical algorithmic knowledge that applies to a given problem.
2.2.4	Compare and contrast alternative solution/methods to select the best methods
3.1.2	Able to identify and document system requirements from stake holders.
3.2.1	Ability to explore design alternatives.
3.2.3	Identify suitable non-functional requirements for evaluation of alternate design solutions.
3.4.2	Ability to implement and integrate the modules
3.4.3	Ability to verify the functionalities and validate the design.
4.1.2	Ability to choose appropriate procedure/algorithm, data set and test cases.
4.2.1	Design and develop appropriate procedures/methodologies based on the study objectives.
5.1.2	Create/adapt/modify/extend tools and techniques to solve engineering problems
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field.
11.3.2	Analyze sourced technical and popular information for feasibility, viability, sustainability, etc.

Course Outcomes:

Learner will be able to

1. *Design and apply the basic concepts and principles of various data types. (LO 1.1, LO 1.2)*
2. *Implement the concepts and operations of linear data structures such as arrays, stacks, queues, and linked lists suing static and dynamic allocation. (LO 2.1, LO 2.2, LO 3.1, LO 3.2, LO 3.3)*
3. *Design and implement various operations and functions of non-linear data structures such as trees and graphs. (LO 4.1, LO 4.2)*
4. *Analyze and compare different techniques of searching, hashing, and sorting for performance efficiency. (LO 5.1, LO 5.2, LO 5.3)*
5. *Apply and evaluate suitable data structures—linked lists, stacks, queues, trees, and graphs—to solve complex computational problems effectively. (LO 6.1, LO 6.2, LO 6.3)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSLBC301.1	3	3	3	3		-	-	-	-	-	-
CSLBC301.2	3	3	3	3		-	-	-	-	-	-
CSLBC301.3	3	3	3	3		-	-	-	-	-	3
CSLBC301.4	3	3	3	3	3	-	-	-	-	-	3
CSLBC301.5	3	3	3	3	3	-	-	-	-	-	3
Average	**3**	**3**	**3**	**3**	**3**	-	-	-	-	-	**3**

Text Books:

1. Data Structures through C in Depth, S. K Srivastava, Deepali Srivastava, 5th Edition, 2011, BPB Publications.
2. Data Structures Using C by Aaron M Tenenbaum, 1st Edition 2018, Pearson India.

3. Data Structures using C, Reema Thareja, 2nd Edition, 2011, Oxford.

Reference Books:

1. Fundamentals of Data Structures, Ellis Horowitz, Sartaj Sahni, 5th Edition, 2010, Galgotia Publications.
2. An introduction to data structures with applications, Jean Paul Tremblay, Paul G. Sorenson; 3rd Edition, 1984, Tata McGraw-Hill.
3. Data Structures using C and C++, Rajesh K. Shukla, 2nd Edition, 2009, Wiley India.

Other Resources:

1. Digital material:
   Web Link- Complete DS Data Structure in one shot | Semester Exam | Hindi (youtube.com)
2. Web Link- https://topperworld.in/dsa-handwritten-notes/

A. CONTINUOUS ASSESSMENT (25 MARKS)

Suggested breakup of distribution

a) Task Execution :10 Marks

Students will be given minimum 12 tasks based on list as per mentioned in the syllabus. Each task carries 10 Marks. Average will be taken of all tasks. Students are expected to execute the program for given task with respect to given data structure, also expected to perform searching & sorting operations on given dataset. Students will be evaluated based on logic building for the given task and expected output.

b) Practical Test: 10 Marks

Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

c) Regularity & Active participation: 05 Marks

B. END SEMESTER EXAMINATION (Practical and Oral Exam) (25 Marks)

Suggested breakup of distribution

a) Task Execution: 10 Marks

Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

b) Results and discussion, Inferences drawn based on above task: 05 Marks

c) Oral based on entire syllabus :10 Marks

Two examiners, one Internal and one External will do the evaluation.

Course Type	Course Code	Course Name	Credits
LBC	CSLBC302	SQL LABORATORY	01

Examination Scheme		
Continuous Assessment	**End Semester Exam (ESE)**	**Total**
25	25	50

Pre-requisite:

1. ESL103 : Programming Laboratory-I (C)

Program Outcomes addressed:

1 PO1: Engineering knowledge
2 PO2: Problem analysis
3 PO3: Design/Development of Solutions
4 PO5: Engineering tool usage
5 PO8: Individual and Collaborative team work
6 PO09: Communication
7 PO11: Life-long learning

Course Objectives:

1. To introduce students database modeling concepts, including entity-relationship (ER) and enhanced entity-relationship (EER) diagrams. Train them to establish relationships, apply integrity constraints, and ensure data consistency using engineering principles
2. Familiarize students with SQL commands for creating, modifying, and securing databases. Teach the use of GRANT, REVOKE, and transaction control commands to enforce access control and optimize database performance.
3. Impart knowledge on implementing views, triggers, and materialized views to automate database operations. Teach structured approaches to enhance database efficiency and performance.
4. To develop designing and implementing real-time database solutions using SQL, functions, procedures, and JDBC connectivity. Develop problem-solving skills by collecting stakeholder requirements and defining problem statements effectively.
5. Foster teamwork by training students to communicate requirements, distribute project tasks, and execute database activities while adhering to deadlines

Module	Details	Hrs.
00.	**Course Introduction** This is foundation course which deals with fundamental concepts of SQL to store, update, remove, search, and retrieve information from the database and it is used in all real time applications for management of data.	**01**
01.	**Introduction to ER and Extended ER diagram**	**07**

	Learning Objective: *Learner is expected to gain knowledge of ER, Extended ER diagram to demonstrate conceptual design of database and expected to implement it* *Learner is expected to define and interpret rules of conversion of ER into relational model to implement a table in a database. Also expected to analyze it for real time application*	
	Contents: **Notation of ER and Extended ER diagram, Mapping the ER and EER Model to the Relational Model** **Task 1:** Identify real world problem and develop the problem statement. Design an Entity-Relationship (ER) / Extended Entity-Relationship (EER) Model ***Learning Outcomes:*** *A learner will be able to* *LO 1.1: Apply engineering fundamentals to evaluate real-world problems and identify clear objectives, and generate suitable ER/EER design solutions that meet functional requirements. (PI- 1.3.1, 2.1.1, 3.2.2,11.1.1,11.2.2)* **Task 2:** Mapping ER/EER to Relational schema model for task 1. ***Learning Outcomes:*** *A learner will be able to* *LO 1.2: Define a clear problem statement, identify required system processes and algorithms, and apply computer science engineering principles to map ER/EER models into an effective relational schema. (PI- 1.4.1, 2.1.2, 3.*1.1, 11.1.1,11.2.2)	
02.	**Types of Language in SQL** ***Learning Objective:*** *Learner is expected to list and identify Data Manipulation commands to implement data manipulation in database for real time application* *Learner is expected to define and apply Data Control commands to assign authorization to the database user as per their role and implement it for real time application* *Learner is expected to acquire knowledge and demonstrate retrieval of information as per requirement of organization and implement it for real time application*	**12**
	Contents: **Data Definition Commands, Data Manipulation commands, Data Control commands, Complex SQL queries** **Task 3:** Create a database using DDL and apply integrity constraints. Learning Outcomes: *A learner will be able to* *LO 2.1: Evaluate problem statements and define clear objectives and scope to identify design constraints and generate suitable database design solutions, including creating a database using DDL and applying integrity constraints that meet functional requirements. (PI- 2.1.1, 2.3.2, 3.1.1, 3.2.2, 11.1.1,11.2.2)* **Task 4:** Perform data manipulations operations on populated database. ***Learning Outcomes:*** *A learner will be able to*	

	LO 2.2: Evaluate problem statements and define clear objectives and scope to identify design constraints and generate suitable solution approaches for performing data manipulation operations on a populated database in order to meet functional requirements. (PI: 2.1.1, 2.3.2, 3.1.1, 3.2.2, 11.1.1,11.2.2) **Task 5:** Perform Authorization using Grant and Revoke. ***Learning Outcomes:*** *A learner will be able to* *LO 2.3: Identify and document authorization requirements, and apply appropriate engineering tools and techniques—including creating or adapting methods as needed—to implement secure database access control using GRANT and REVOKE. (PI: 3.1.2, 5.1.1, 5.1.2, 5.2.2, 11.1.1,11.2.2).* **Task 6:** Implement Basic and complex SQL queries ***Learning Outcomes:*** *A learner will be able to* *LO 2.4: Identify and document query requirements, select appropriate engineering tools and techniques, and adapt or apply them effectively to implement both basic and complex SQL queries for solving database problems. (PI: 3.1.2, 5.1.1, 5.1.2, 5.2.2, 11.1.1,11.2.2)*	
03.	**Views and Triggers** *Learner is expected to define and demonstrate Views and Triggers to provide access to the required data only and implement it for real time application.* **Contents:** **Views and Triggers** **Task 7 :** **Implementation of Views and Triggers real time application** ***Learning Outcomes:*** *A learner will be able to* *LO 3.1: Evaluate real-world application requirements and define a precise problem statement to identify constraints, document stakeholder needs, and apply engineering fundamentals and computer science principles to design, implement, and adapt views and triggers using appropriate modern tools for effective database solutions.(PI: 1.3.1, 1.4.1, 2.1.1, 2.3.2, 3.1.1, 3.1.2, 5.1.1, 5.1.2, 11.1.1,11.2.2)*	**02**
04.	**TCL and Functions and procedures** ***Learning Objectives:*** ***Learner is expected to define and demonstrate Views and Triggers to provide access to the required data only and implement it for real time application. Learner is expected to define and demonstrate Transaction Control Language to use commit and rollback command to save tables in databases and implement it for real time application*** **Contents:** **Transaction control Language, Functions and procedures, Database Programming , Mini project on real time problem** **Task 8:** Execute TCL commands for real time application ***Learning Outcomes:*** *A learner will be able to*	**07**

	LO 4.1: Apply engineering fundamentals and computer science principles to solve real-time database problems by identifying and using appropriate tools and adapting techniques to execute TCL commands effectively. (PI- 1.4.1, 1.3.1, 5.1.1, 5.1.2, 11.1.1,11.2.2) **Task 9:** Implement functions and procedures in SQL ***Learning Outcomes:*** *A learner will be able to* *LO 4.2: Define clear problem objectives, identify stakeholder requirements and design constraints, interpret relevant technical information, and collaboratively generate well-structured solutions by implementing SQL functions and procedures while staying updated with evolving database technologies. (PI- 3.1.2, 3.1.1, 11.2.2, 11.1.1, 2.3.2, 9.1.1, 8.1.1, 2.1.1, 8.3.1, 9.3.1, 11.1.1,11.2.2)* **Task 10:** Demonstrate database connectivity for real time application ***Learning Outcomes:*** *A learner will be able to* *LO 4.3 Evaluate real-time application requirements, identify design constraints, and apply engineering fundamentals and computer science principles to design and demonstrate database connectivity by interpreting technical information, generating and selecting effective solution approaches, collaborating efficiently in teams, and staying updated with evolving technologies. (PI- 2.1.1, 2.3.2,9.1.1, 9.3.1, 8.1.1, 8.3.1, 1.3.1, 1.4.1, 2.1.1, 2.3.2, 3.3.3, 3.2.2, 11.1.1, 11.2.2)*	
	Course Conclusion	**01**
	Total	**30**

Performance Indicators:

P.I. No.	**P.I. Statement**
1.3.1	Apply engineering fundamentals
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem
2.1.1	Evaluate problem statements and identifies objectives
2.1.2	Identifies processes/modules/algorithms of a computer based system and parameters to solve a problem
2.3.2	Identify design constraints for required performance criteria.
3.1.1	Able to define a precise problem statement with objectives and scope
3.1.2	Able to identify and document system requirements from stake holders
3.2.2	Ability to produce a variety of potential design solutions suited to meet functional requirements
5.1.1	Identify modern engineering tools, techniques and resources for engineering activities
5.1.2	Create/adapt/modify/extend tools and techniques to solve engineering problems
5.2.2	Demonstrate proficiency in using discipline specific tools
8.1.1	Recognize a variety of working and learning preferences; appreciate the value of diversity on a team

8.3.1 Present results as a team, with smooth integration of contributions from all individual efforts

9.1.1 Read, understand and interpret technical and nontechnical information

9.3.1 Create engineering-standard figures, reports and drawings to complement writing and presentations

11.1.1 Describe the rationale for requirement for continuing professional development

11.2.2 Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field

Course Outcomes: A learner will be able to -

1. Apply engineering and computer science principles to create well-structured database models, ensuring proper entity relationships, normalization, and integrity constraints. *(LO 1.1, 1.2)*
2. Create, modify, and secure databases using SQL commands, enforcing constraints, executing queries, and applying transaction control mechanisms. *(LO 2.1, 2.2, 2.3,4.1)*
3. Implement views and triggers to automate database operations and define materialized views to optimize data retrieval and storage. *(LO 3.1)*
4. Design and implement real-time databases using SQL functions, stored procedures, and JDBC connectivity while considering stakeholder requirements. *(LO 4.1.4.2)*
5. Communicate project requirements, distribute tasks efficiently, execute database-related activities as per the project plan, and meet deadlines. *(LO 4.3)*

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSLBC302 .1	3	3	3	-	-	-	-	-	-	-	3
CSLBC302 .2	3	3	3	-	3	-	-	-	-	-	3
CSLBC302.3	3	3	3	-	3	-	-	-	-	-	3
CSLBC302.4	3	3	-	-	-	-	-	3	3	-	3
CSLBC302.5	-	3	3	-	-	-	-	3	3	-	3
Average	**3**	**3**	**3**	-	**3**	-	-	**3**	**3**	**-**	**3**

Text Books :

1. Database System Concepts, Korth, Slberchatz, Sudarshan , Sixth Edition, 2011, McGraw Hill Education
2. Fundamentals of Database Systems, Elmasri and Navathe, Sixth Edition, 2010, Pearson
3. Database Management Systems, Raghu Ramkrishnan and Johannes Gehrke, ISE Editions, 1997 ,McGraw-Hill Education

Reference Books :

1. Database Systems Design, Implementation and Management , Peter Rob and Carlos Coronel, 9th Edition, 2009, Thomson Learning
2. SQL & PL / SQL for Oracle 11g Black Book, Dr. P. S. Deshpande, ISE Edition, 2011,Dreamtech Press

3. Database Management Systems , G. K. Gupta , 1st edition,2011,McGraw – Hill

Other Resources :

1. NPTEL Course on Database Management system by Prof.P.P.Das, Department of Computer Science and Engineering, IIT Khargpur
   Web link- https://archive.nptel.ac.in/courses/106/105/106105175/

2. Coursera course on Introduction to Relational Databases
   Web link- https://www.coursera.org/learn/introduction-to-relational-databases#modules

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (25 Marks)

Suggested breakup of distribution

a) Task Execution :10 Marks

Students will be given 10 tasks based on list as per mentioned in the syllabus. Each task carries 10 Marks. Average will be taken of all tasks. Students are expected to formulate problem statement, create relational schema, implement relational schema by using constraints and retrieve required information. Students will be evaluated based on logic building for the selected problem, rectification of syntax errors and expected output.

Students will be evaluated based on following:

i. Problem Understanding Correctly interprets given SQL task (tables, constraints, requirements). (02 marks)
ii. Query Accuracy SQL query is syntactically correct and produces correct output (02 marks)
iii. Appropriate use of DDL/DML/Constraints/Joins/Functions/Clauses as required. (02 marks)
iv. Execution & Output Executes without errors and displays expected results (02 marks)
V. Regularity in performance and submission of task (02 marks)

b) Mini Project Evaluation: 10 Marks

A group of 3-4 students should be assigned a real life problem statement. Students are expected to research and collect required resources to create a frontend and backend for the selected problem. Students should prepare a presentation/problem solving of 10-15 minutes. This project should be graded for 10 marks depending on the parameters as analysis, design, conversion to relational schema, database creation and implementation for selected problem statement.

Students will be evaluated based on following:

i Problem Understanding Correctly interprets given SQL task (tables, constraints, requirements). (02 marks)

ii SQL query is syntactically correct and produces correct output (02 marks)

iiiAppropriate use of DDL/DML/Constraints/Joins/Functions/Clauses as required. (02 marks)

iv Execution & Output Executes without errors and displays expected results (02 marks)

v. Regularity in performance and submission of task (02 marks)

c) Regularity & Active participation: 05 Marks

B. END SEMESTER EXAMINATION (25 MARKS)

a) Task Execution: 10 Marks
   Students will be given task (different task for every student) and will be evaluated as per the parameters mentioned in continuous assessment.
b) Results and discussion, Inferences drawn from the above task: 05 Marks
c) Oral based on entire syllabus :10 Marks

Two examiners, one Internal and one External will do the evaluation.

Course Type	Course Code	Course Name	Credits
SBL	CSSBL301	PYTHON LABORATORY	02

Examination Scheme		
Continuous Assessment	**End Semester Exam (ESE)**	**Total**
50	50	100

Pre-requisite :

1. ESL103- Programming Laboratory-I (C)
2. ESL205- Programming Laboratory-II (Java)

Program Outcomes addressed :

1. PO 1: Engineering knowledge
2. PO 2: Problem analysis
3. PO 3: Design/Development of Solutions
4. PO 5: Engineering tool usage
5. PO 8: Individual and Collaborative team work
6. PO 9: Communication

Course Objectives :

1. To provide basics of python programming including data types, operator, conditional statements, looping statements, input, and output functions in Python
2. To impart knowledge of advanced data types of python programming.
3. To inculcate knowledge of object oriented programming concepts in python.
4. To impart with concepts of modules, packages, multithreading and exception handling.
5. To acquaint learner with File handling, GUI & database programming.
6. To impart knowledge of data visualization using Matplotlib, Data analysis using Pandas and Web programming using Flask/Streamlit.

Module	Detailed Contents	Hrs
00.	**Course Introduction** This course provides programming concepts of python from basics to advanced concepts. Design and write fully functional Python programs using commonly used data structures, custom functions, and reading and writing to files overview of oops concepts. provides basics of creating modules, packages and importing packages and provides fundamentals of GUI designing, database connectivity and web programming, data visualization and analysis. These concepts are essential for developing skill sets to work in the field of data science.	**01**
01.	**Basics of python and introduction to advanced data types, strings and special functions .**	**10-12**
	Learning Objective/s:	

	Learner is expected to infer and apply regular programming constructs to write python programs ,use advanced data types in python for defining an ordered and unordered collections and special functions for code optimization.	
	Contents: Basics of Python: Variables, operators, control structures, conditional structures, functions, lists, tuples, set, dictionary and its built-in functions strings and its built in functions, map reduce and filter function. **Task 1:** Write a program to identify and use variables, operators control and conditional structures for a given task. **Task 2:** Write a program to modularize task1 using functions for a given task. **Task 3:** Identify and use advanced data type to store information of same type or different type and perform built in operations and also analyze the output for any 2 advanced data types identified for task2 **Task4:** Use special functions map, reduce and filter on task3 to achieve code optimization. **Task 5:**Implement string built in functions on task4 to search a given string**.**	
	Self-Learning Topics: *Decorators, iterators, generator functions* ***Learning Outcomes :*** *A learner will be able to* *LO 1.1: Use variables, operators, control and conditional structures to write program in python. (P.I.-1.3.1)* *LO 1.2: Use functions in python to modularize a given program with set of statements. (P.I.-1.4.1)* *LO 1.3: Identify and use advanced data types and their built in operations that applies to a given problem. (P.I.-2.1.2)* *LO 1.4: Use and analyze working of any 2 advanced data types that applies to a given problem. (P.I.-2.1.2)* *LO 1.5: Implement string built in functions to meet functional requirements to write a program in python. (P.I.-3.2.2)* *LO 1.6: Apply Map and reduce functions to solve a given program in python as a design alternative for for loop. (P.I.-3.2.1)*	
02.	**Object oriented programming concepts in python** ***Learning Objective/s:*** *Learner is expected to summarize and apply basic concepts of object-oriented programming in python to make code more reusable and easier to work with larger programs.*	**11-13**
	Contents: Overview of Object -oriented programming, creating classes and objects, self-Variable, constructors, operator overloading, Inheritance: Types of Inheritance (Single, Multiple, Multi -level, Hierarchical) constructors in inheritance, method overriding, abstract class, abstract methods.	

	Task 6: Create a class for the task 5 add variables, methods, contructors to the class and access the members of the class. **Task 7:** Create a child class for task6 and inherit properties from task6 to achieve code reusability. **Task 8:** Create same functions in parent and child class to demonstrate method overriding for task 7. **Task 9:** Create abstract class add abstract methods to it and implement methods in child class for task 7.	
	Self-Learning Topics: *Single Responsibility Principle (SRP), Open/Closed Principle (OCP).*	
	Learning Outcomes: *A learner will be able to* *LO 2.1: Create a class and declare member variables, add constructors to initialize variables (P.I.-1.3.1)* *LO 2.2: Use concepts of inheritance to achieve code reusability for solving a problem of code maintainability. (P.I -1.4.1).* *LO 2.3: Apply method overriding to implement and integrate the modules. (P.I - 3.2.2)* *LO 2.4: Identify and use simplification using abstraction . (P.I - 3.2.1)*	
03.	**Introduction to modules, packages, exception handling multithreading in python.** ***Learning Objective/s:*** *Learner is expected to paraphrase and implement the concepts of user defined and built in modules and packages that are useful for code modularization, code reusability and use multi-threading and exception handling in python..*	**11-13**
	Contents: Modules: Writing modules, importing objects from modules, python built -in modules Numpy and Scipy Packages: creating user defined packages and importing packages. Multi -threading: creating threads in python, exception handling: Compile time errors, Runtime errors, exceptions, types of exception, try statement, except block, raise statement, assert statement, user -defined exceptions. **Task 10:** Create a user defined package and all the above tasks as modules to the package and import any one module in another task to achieve code reusability. **Task 11:** Create threads for any two functions of task3 to demonstrate multithreading. **Task 12:** Use try except block to handle name error,value error and type error for any of the above task.	
	Self-Learning Topics: *Study of media related modules in python.*	

	Learning Outcomes : *A learner will be able to* *LO 3.1: Identify and write user defined exception for a given problem. (P.I -2.1.2).* *LO 3.2: Identify and use functions of NumPy and SciPy module to perform matrix operations and solve algebraic equations. (P.I -2.2.3)* *LO 3.3: Create your own module and implement it in another file. (P.I -3.2.1)* *LO3.4: Implement user defined packages to achieve code modularity in programming. (P.I-3.4.2).*	
04.	**File handling and designing graphical user interface in python** ***Learning Objective/s:*** *Learner is expected to summarize and implement basics of File handling in Python.* *Learner is expected to summarize and create backend databases and connect GUI with databases to perform CRUD operations.*	**9-11**
	Contents: File Handling operations in python, Graphical user interface (GUI): different GUI tools in python (Tkinter, PyQt) Working with containers, Canvas, Frame, Widgets (Button, Label, Text, Scrollbar, Check button, Radio button, Entry, Spinbox, Message etc.) Connecting GUI with databases to perform CRUD operations. (on supported databases like SQLite, MySQL, PostgreSQL) **Task 13:** Write a program to perform file handling operations for task12. **Task 14:** Design GUI using TKinter for task2. **Task 15:** Create a database for task 14 and connect it to the GUI designed and perform CRUD operations.	
	Self-Learning Topics: *Design GUI using Kivy, MongoDB database*	
	Learning Outcomes : *A learner will be able to* *LO 4.1: Use basics of file handling operations in python. (P.I -1.3.1)* *LO 4.2: Use TKinter to design GUI for a given problem. (P.I.-1.4.1)* *LO 4.3: Create a database using SQLite/MySQL according to the given problem statement. (P.I-3.2.1)* *LO 4.4: Perform and integrate CRUD operations in python. (P.I -3.4.2)* *LO 4.5: Identify tools to install SQLite, MySQL. (P.I -5.1.1)* *LO 4.6: Use tools to install SQLite, MySQL and perform CRUD operations. (P.I -5.1.2)*	
05.	**Data Visualization, analysis and web programming in python** ***Learning Objective/s:*** *Learner is expected to infer concepts of data visualization and analysis of data using matplotlib and pandas.* *Learner is expected infer skills of web programming using Flask/Django framework.*	**13-15**

	Content: Visualization using Matplotlib: working with plots (line plot, bar graph, histogram, scatter plot, area plot, pie chart etc.), working with multiple figures. Data manipulation and analysis using Pandas: Introduction to Pandas, importing data into Python, series, data frames, indexing data frames, basic operations with data frame, filtering, combining and merging data frames, removing duplicates. Web programming using Flask/Streamlit Task 16: Write a program to plot graph for visualizing output for any one task according to the problem statement. Task 17: Write a program to create a simple pandas data frame for task3. Task 18: Write a program to read a csv file containing data related to task2 clean and analyze the data according to the problem statement. Task 19: Develop a web application using flask/Streamlit by integrating above task according to the problem statement.	
	Self-Learning Topics: *API essentials in python, pygame tool*	
	Learning Outcomes: *A learner will be able to* *LO 5.1: Identify which graph to plot using matplotlib for a given problem. (P.I -2.1.2)* *LO 5.2: Use matplotlib library to plot various graphs for analyzing data visually according to the given problem (P.I -2.4.2)* *LO 5.3: Design and demonstrate web application using flask for a given task. (P.I -3.1.1,8.1.1,8.2.1,9.1.2,9.1.3)* *LO 5.4: Identify and use various ways of cleaning data using pandas. (P.I -3.2.2)* *LO 5.5: Identify tools to install matplotlib and pandas. (P.I -5.1.1)* *LO 5.6: Use tool to install matplotlib, pandas module and flask environment. (P.I -5.1.2).*	
	Total	**60**

Performance Indicators:

P.I. No.	P.I. Statement
1.3.1	Apply engineering fundamentals
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem
2.1.2	Identifies processes/modules/algorithms of a computer-based system and parameters to solve a problem
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions
2.4.2	Analyze and interpret the results using contemporary tools.
3.1.1	Able to define a precise problem statement with objectives and scope.
3.2.1	Ability to explore design alternatives.
3.2.2	Ability to produce a variety of potential design solutions suited to meet functional requirements.
3.4.2	Ability to implement and integrate the modules.
5.1.1	Identify modern engineering tools, techniques and resources for engineering activities .

5.1.2 Create/adapt/modify/extend tools and techniques to solve engineering problems.

8.1.1 Recognize a variety of working and learning preferences; appreciate the value of diversity on a team

8.2.1 Demonstrate effective communication, problem solving, conflict resolution and leadership skills

9.1.2 Produce clear, well-constructed, and well-supported written engineering documents

9.1.3 Create flow in a document or presentation - a logical progression of ideas so that the main point is clear

Course Outcomes:

Learner will be able to

1. Apply basic concepts of structure, syntax, and semantics of the Python language and use advanced data types and special functions in python. *(LO 1.1,LO 1.2, LO 1.3,LO 1.4, LO 1.5,LO 1.6)*
2. Apply the concepts of object-oriented programming concepts in Python. *(LO 2.1,LO 2.2, LO 2.3,LO 2.4)*
3. Create Python applications using modules, packages and use multithreading and exception handling.(*LO 3.1,LO 3.2, LO 3.3,LO 3.4)*
4. Implement File Handling operations, create GUI applications and perform database operations in Python.*(LO 4.1,LO 4.2, LO 4.3,LO 4.4, LO 4.5,LO 4.6)*
5. Develop robust applications using the latest Python trends and technologies. (*LO 5.1, LO 5.2, LO 5.3, LO 5.4, LO 5.5, LO 5.6)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSSBL301.1	3	3	3	-	-	-	-	-	-	-	-
CSSBL301.2	3	3	3	-	-	-	-	-	-	-	-
CSSBL301.3	3	3	-	-	3	-	-	-	-	-	-
CSSBL301.4	3	3	-	-	3	-	-	-	-	-	-
CSSBL301.5	3	3	3	-	3	-	-	3	3	-	-
Average	**3**	**3**	**3**	-	**3**	**-**	**-**	3	**3**	-	**-**

Text Books:

1. Introduction to computing and problem-solving using python, E Balagurusamy, First Edition,2017, McGraw Hill Publication.
2. Python Programming Using Problem Solving Approach, Reema Thareja, Second Edition, 2023, Oxford University. Press.
3. Python: The Complete Reference, Martin C. Brown, Fourth Edition, 2018, McGraw-Hill Publication.

Reference Books:

1. Headfirst Python- A Brain Friendly Guide, Paul Barry, 2nd Edition,2016, O'Reilly Media, Inc.
2. Problem Solving and Python Programming, Arockia Mary P, Kindle Edition,2021, Shanlax Publications.

Other Resources :

1. NPTEL Course: The Joy of Computing using Python, By Prof. Sudarshan Iyengar, Department of Computer Science and Engineering, IIT Ropar :-Web link- https://nptel.ac.in/courses/106106182/
2. NPTEL Course: Python for Data science By Prof. Ragunathan Rengasamy, Department of Computer Science and Engineering, IIT Madras :-Web link- https://archive.nptel.ac.in/courses/106/106/106106212/
3. Web link-https://www.w3schools.com/python/
4. Web link-https://www.geeksforgeeks.org/python-programming-language/
5. Web link-https://www.javatpoint.com/python-tutorial

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Task Execution (30 Marks)

Students will be given minimum 10 tasks.

Students are expected to

i. Identify variables, data types methods/approach required to write the code for the given task and apply the same.
ii. Execute given task for different inputs and verify the result
iii. Import different Python libraries to solve given problem
iv. Implement basic file handling in Python
v. Create backend databases and connect GUI to perform CRUD operations
vi. Use data visualization tools
vii. Apply appropriate mechanisms to handle unexpected errors.

Students will be evaluated based on following:

i. Logic building for the given task (10 marks)
ii. Rectifying logical errors and syntax errors (06 marks)
iii. Well-structured and organized program (06marks)
iv. Verification of experiment output for different inputs (04 marks)
v. Regularity in performance and submission of task (04 marks)

Refer the sample task given below.

Example:

Create a Teacher class to add details of the teachers their subject to respective class and display details using methods.

Students are expected to identify.

i. variables, data types methods/approach required to create teacher class and add methods to display details of a given teacher
ii. Execute given task for different inputs and verify the result
iii. Follow the coding standards
iv. Identify errors and rectify the errors.

Students will be evaluated based on following:

i. Logic building for the given task (10 marks)
ii. Rectifying logical errors and syntax errors (06 marks)
iii. Well-structured and organized program (06 marks)
iv. Verification of experiment output for different inputs (04 marks)
v. Regularity in performance and submission of task (04 marks)

2. Regularity and active Participation (05 Marks)

3. Practical Test (15 Marks)

a) Task Execution: 10 Marks

Students will be given task (different task for every student) to execute and will be evaluated as per the parameters mentioned in continuous evaluation

Students will be evaluated based on following:

i. Logic building for the given task (04 marks)
ii. Rectifying logical errors and syntax errors (02 marks)
iii. Well-structured and organized program (02 marks)
iv. Verification of experiment output for different inputs (02 marks)

b) Oral based on covered syllabus: 05 Marks

B. END SEMESTER EXAMINATION (Practical & Oral Exam) (50 Marks)

a) Task Execution: 30 Marks
Students will be given task (different task for every student) to execute and will be evaluated as per the parameters mentioned below
i. Logic building for the given task (10 marks)
ii. Rectifying logical errors and syntax errors (06 marks)
iii. Well-structured and organized program (06 marks)
iv. Verification of experiment output for different inputs (08 marks)

b) Presentation of Results and conclusion, Inferences drawn: 05 Marks

c) Oral based on entire syllabus: 15 Marks

Two examiners, one Internal and one External will do the evaluation.

Course Type	Course Code	Course Name	Credits
MNP	CSMNP301	MINI PROJECT- 1A	01

Examination Scheme		
Continuous Assessment	**Practical /Oral**	**Total**
50	-	50

Program Outcomes addressed:

1. PO1 : Engineering knowledge
2. PO2 : Problem Analysis
3. PO3 : Design/Development of Solutions
4. PO4 : Conduct investigations of complex problems
5. PO5 : Engineering Tool Usage
6. PO6 : The Engineer & The World
7. PO7 : Ethics
8. PO8 : Individual & Collaborative team work
9. PO9 : Communication
10. PO10 : Project Management & Finance
11. PO11 : Life-long learning

Course Objectives:

1. To familiarize students about available infrastructure at Department/Institute level, online resour plagiarism, expectations from MP 1A and 1B, etc.
2. To guide students in identifying societal or research needs and formulating them into problem statements.
3. To facilitate problem-solving in group settings.
4. To apply basic engineering principles to address identified problems.
5. To foster self-learning and research skills.

Guidelines for the Mini Project

- At the beginning of semester-III, project guides are required to conduct around 4 hours' orientation sessions including following topics:
  - Familiarizing students about infrastructure available at Department/Institute level and how to use it.
  - How to identify societal problems and formulate project problem statement.
  - How to carry out literature survey.
  - What is plagiarism and what care needs to be taken while writing a report.

- What is project report template and how it should be used.
- What are expectations from mini-projects 1A and 1B.

• Mini project may be carried out in one or more form of following:

- Product preparations, prototype development model, fabrication of set-ups, laboratory experiment development, process modification/development, simulation, software development, integration of software (frontend-backend) and hardware, statistical data analysis, creating awareness in society/environment etc.

• Students must form groups of 3 to 4 members either from the same or from different departments.

• Groups should conduct surveys to identify needs and develop problem statements in consultation with faculty.

• An implementation plan in Gantt/PERT/CPM chart format covering weekly activities must be submitted.

• Each group must maintain a logbook to record weekly progress, to be verified by the faculty supervisor.

• Faculty input should emphasize guiding by faculty and self-learning by group members.

• Groups should propose multiple solutions, select the best one in consultation with the supervisor, and develop a working model.

• The solution to be validated with proper justification and report to be compiled in standard format of the Institute. Research papers, competition certificates may be submitted as part of annexure to the report.

• With the focus on self-learning, innovation, addressing societal/research/innovation problems and entrepreneurship quality development within the students through the Mini Projects, it is preferable that a single project of appropriate level and quality be carried out in two semesters by all the groups of the students.

• However, based on the individual students or group capability, with the mentor's recommendations, if the proposed Mini Project adhering to the qualitative aspects mentioned above, gets completed in odd semester, then that group can be allowed to work on the extension of the Mini Project with suitable improvements/modifications or a completely new project idea in even semester. This policy can be adopted on a case by case basis.

Course Outcomes:

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

1. Identify problems based on societal or research needs and methodology for solving them.

2. Apply knowledge and skills to solve societal problems collaboratively.
3. Develop interpersonal skills necessary for teamwork.
4. Analyze, verify, and validate results effectively through various methodologies, including, test cases/benchmark data/theoretical/inferences/experiments/simulations, etc.
5. Evaluate the societal and environmental impacts of proposed solutions.
6. Adhere to standard engineering practices.
7. Excel in written and oral communication by technical report writing, oral presentation, and publishing results in
    - Research/white paper/article/blog writing/publication, etc.
    - Business plan for entrepreneurship product creation
    - Patent filing/copyright.
8. Gain technical competencies by participating in competitions, hackathons, etc.
9. Demonstrate lifelong learning capabilities through self-directed group projects.
10. Apply project management principles effectively.

In-Semester Continuous Assessment Guidelines:

- The Head of the Departments will assign a guide to each of the mini-projects and shall form a progress monitoring committee. The guide will carry out weekly monitoring of the project's progress. The committee shall carry out in-semester project evaluation based on presentations with a minimum of two evaluations per semester.
- Assessment will be based on individual contributions, understanding, and responses to questions asked.
- Continuous Assessment marks distribution in semester III (50 marks):
    - 05 marks for the Topic Approval Presentation in front of the progress monitoring committee
    - 15 marks for the Mid-Semester Progress Presentation in front of the progress monitoring committee
    - 25 marks for the Final Report & Presentation
    - 05 marks for Regularity and active participation

The review/progress monitoring committee will assess projects based on the following criteria.

Semester III:

- Theoretical solution completion, including component/system selection/design of software solution and cost analysis.

- Two reviews will occur:
  - The first review will focus on finalizing the problem statement (topic approval).
  - The second review will focus on finalizing the proposed solution.

In addition to the above-mentioned points, the following performance criteria shall be included during the in-semester continuous assessment:

1. Quality of survey and need identification.
2. Clarity and innovativeness in problem definition and solutions.
3. Requirement gathering feasibility study, cost-effectiveness, and societal impact of proposed solutions.
4. Completeness and full functioning of the working model.
5. Effective use of skill sets and engineering norms.
6. Verification & validation of the solutions/test cases.
7. Individual contributions to the group.
8. Clarity in written and oral communication.
9. Participation in technical paper presentations/project competitions/hackathon competitions, etc.

Course Type	Course Code	Course Name	Credits
HSS	HSS301	PRODUCT DESIGN	02

Examination Scheme		
Continuous Assessment	**End Semester Examination**	**Total**
50	-	50

Program Outcomes addressed:

1. PO2 Problem analysis
2. PO3 Design/Development of solutions
3. PO5 Engineering tool usage
4. PO6 The engineer and the world
5. PO7 Ethics
6. PO8 Individual and collaborative team work
7. PO10 Project management & finance
8. PO11 Life-long learning

Course Objectives:

1. Understand the product design process and its user-centered principles.
2. Apply fundamental design principles to create innovative product designs.
3. Demonstrate proficiency in generating and evaluating design concepts through ideation techniques.
4. Evaluate and synthesize sustainable and user-centric design practices in product development.

Module	Details	Hrs.
00.	**Course Introduction** The course Product Design provides a thorough understanding of the principles, methods, and methodologies used in developing unique and functional products. Whether you want to be an industrial designer, a UX/UI specialist, or a creative problem solver, this course will provide you the necessary knowledge and abilities to envision, develop, and revise products that fulfill user needs and market demands. Students will investigate many areas of product design, such as aesthetics, usability, and sustainability, using both theory and hands-on practice. Design thinking, prototyping, and research will demonstrate to students how to turn ideas into concrete things that improve user experience and address real-world challenges.	**01**
01.	**Introduction to Product Design** ***Learning Objective:*** *Understand the fundamental principles and key elements that contribute to effective product design.*	**03-05**
	Contents: Overview of product design process, Importance of user-centered design, Design thinking methodologies, Case studies of successful product	

	designs, Introduction to design tools and software (e.g., Sketch, Adobe XD)	
	Learning Outcomes: *A learner will be able to* *LO 1.1: Apply design thinking methodologies to develop user-centered solutions.* (*P.I.- 2.1.1, 2.3.1, 3.2.1, 3.3.1*) *LO 1.2: Gain introductory experience with digital design tools.* (*P.I. – 5.1.1, 5.2.1*)	
02.	**Design Principles and Fundamentals** ***Learning Objective:*** *Understand and apply core design principles to create functional and aesthetically pleasing products.*	**05-07**
	Contents: Understanding design principles (e.g., balance, hierarchy, contrast), Human factors in design (ergonomics, anthropometrics), Material selection and properties, Basics of aesthetics and styling, Hands-on exercises in sketching and prototyping	
	Learning Outcomes: *A learner will be able to* *LO 2.1: Apply fundamental design principles such as balance, contrast, proportion, and harmony to create aesthetically and functionally effective designs.* (*P.I. – 3.1.5, 3.2.3, 6.1.1*) *LO 2.2: Analyze and evaluate design elements to develop user-centered solutions that enhance usability, accessibility, and overall user experience.* (*P.I. – 3.1.1, 6.2.1*)	
03.	**Concept Generation and Ideation** ***Learning Objective:*** *Develop creative ideas and transform them into viable product concepts through structured ideation techniques.*	**05-07**
	Contents: Techniques for brainstorming and idea generation, Sketching and visualization techniques, Developing design briefs and specifications, Evaluating and selecting design concepts, Design for X, Rapid prototyping methods (e.g., 3D printing, CNC machining)	
	Learning Outcomes: *A learner will be able to* *LO 3.1: Apply various ideation techniques such as brainstorming, mind mapping, and SCAMPER to generate innovative and creative product concepts*. (*P.I. – 2.4.4, 3.1.6, 3.2.1*) *LO 3.2: Develop and evaluate multiple design concepts based on user needs, feasibility, and functionality to refine ideas into viable solutions.* (*P.I. – 2.2.4, 3.1.1, 3.1.6*)	
04.	**Product Lifecycle** ***Learning Objectives:*** *Understand the stages of a product's lifecycle and how they influence design, development, and sustainability.*	**03-05**
	Contents: Detailed overview of the product development lifecycle, Cost estimation	

	and budgeting, Marketing and Market research, Regulatory and compliance requirements (e.g., safety standards)	
	Learning Outcomes: *A learner will be able to* *LO 4.1: Understand and analyze the stages of the product lifecycle and their impact on design, marketing, and sustainability decisions.* (*P.I. – 3.1.1, 3.1.6, 6.3.2, 11.3.2*) *LO 4.2: Analyze the influence of lifecycle considerations such as material selection, manufacturing processes, and end-of-life disposal to develop sustainable and cost-effective product solutions.* (*P.I. – 3.1.5, 6.3.1, 6.4.1, 6.4.2, 11.1.1, 11.2.2*)	
05.	**User Experience (UX) Design** ***Learning Objective:*** *Design intuitive and user-friendly products by applying UX principles and usability testing.*	**03-05**
	Contents: Understanding user needs and behaviour, Usability testing and feedback gathering, Wire-framing and prototyping for digital products, Iterative design process, Accessibility and inclusive design principles	
	Learning Outcomes : *A learner will be able to* *LO 5.1: Apply UX design principles such as usability, accessibility, and interaction design to create intuitive and user-friendly products.* (*P.I. – 3.1.6, 3.3.1, 5.2.2*) *LO 5.2: Conduct user research and usability testing to analyze user needs, gather feedback, and refine designs for an enhanced user experience.* (*P.I. – 3.1.1, 3.1.6, 5.1.2, 5.2.1, 10.3.1, 10.3.2*)	
06.	**Sustainability in Product Design** ***Learning Objective:*** *Incorporate sustainable practices and materials to create environmentally responsible product designs.*	**03-05**
	Contents: Environmental impact assessment in product design, Sustainable materials and manufacturing processes, Design for disassembly and recycling, Circular economy principles Case studies of eco-friendly product designs.	
	Learning Outcomes: *A learner will be able to* *LO 6.1: Apply sustainable design principles by selecting eco-friendly materials, optimizing manufacturing processes, and minimizing environmental impact throughout the product lifecycle.* (*P.I. – 3.1.5, 6.3.2, 7.1.1, 11.3.1*) *LO 6.2: Assess the lifecycle impact of products in terms of resource consumption, carbon footprint, and end-of-life disposal to develop eco-friendly and socially responsible design solutions.* (*P.I. – 3.4.1, 6.4.1, 7.2.2, 11.3.2*) *LO 6.3: Demonstrate good communication and collaboration with interdisciplinary teams by incorporating sustainable design concepts, explaining environmental and social implications, and enabling cross-disciplinary discussions to create new, eco-friendly product solutions.* (*P.I. – 8.2.1, 8.3.1*)	
	Course Conclusion	

		01
	Total	**30**

Performance Indicators:

P.I. No.	P.I. Statement
2.1.1	Articulate problem statements and identify objectives.
2.2.4	Compare and contrast alternative solution processes to select the best process.
2.3.1	Combine scientific principles and engineering concepts to formulate model/s (mathematical or otherwise) of a system or process that is appropriate in terms of applicability and required accuracy.
2.4.4	Extract desired understanding and conclusions consistent with objectives and limitations of the analysis.
3.1.1	Recognize that need analysis is key to good problem definition.
3.1.5	Explore and synthesize engineering requirements considering health, safety risks, environmental, cultural and societal issues.
3.1.6	Determine design objectives, functional requirements and arrive at specifications.
3.2.1	Apply formal idea generation tools to develop multiple engineering design solutions.
3.3.1	Apply formal decision-making tools to select optimal engineering design solutions for further development.
3.2.3	Identify suitable criteria for the evaluation of alternate design solutions.
3.4.1	Refine a conceptual design into a detailed design within the existing constraints (of the resources)
5.1.1	Identify modern engineering tools such as computer-aided drafting, 46odelling and analysis; techniques and resources for engineering activities.
5.1.2	Create/adapt/modify/extend tools and techniques to solve engineering problems.
5.2.1	Identify the strengths and limitations of tools for (i) acquiring information, (ii) 46odelling and simulating, (iii) monitoring system performance, and (iv) creating engineering designs.
5.2.2	Demonstrate proficiency in using discipline-specific tools.
6.1.1	Identify and describe various engineering roles; particularly as pertains to protection of the public and public interest at the global, regional and local level.
6.2.1	Interpret legislation, regulations, codes, and standards relevant to your discipline and explain its contribution to the protection of the public.
6.3.1	Identify risks/impacts in the life-cycle of an engineering product or activity.
6.3.2	Understand the relationship between the technical, socio-economic and environmental dimensions of sustainability.
6.4.1	Describe management techniques for sustainable development.
6.4.2	Apply principles of preventive engineering and sustainable development to an engineering activity or product relevant to the discipline.
7.1.1	Identify situations of unethical professional conduct and propose ethical alternatives.
7.2.2	Examine and apply moral & ethical principles to known case studies.
8.2.1	Demonstrate effective communication, problem-solving, conflict resolution and leadership skills.
8.3.1	Present results as a team, with smooth integration of contributions from all individual efforts.
10.3.1	Identify the tasks required to complete an engineering activity, and the resources required to complete the tasks.
10.3.2	Use project management tools to schedule an engineering project, so it is completed on time and on budget.
11.1.1	Describe the rationale for the requirement for continuing professional development.
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field.
11.3.1	Source and comprehend technical literature and other credible sources of information.
11.3.2	Analyze sourced technical and popular information for feasibility, viability, sustainability, etc.

Course Outcomes: A learner will be able to –

1. Apply design thinking methodologies effectively to solve design problems. (*LO 1.1, LO 2.1, LO 2.2, LO 3.1, LO 3.2*)
2. Demonstrate proficiency in utilizing design tools and techniques for product development. (*LO 1.2, LO 5.1*)
3. Communicate and collaborate effectively for interdisciplinary teamwork. (*LO 6.3*)
4. Create functional and aesthetically pleasing product designs. (*LO 5.2*)

5. Integrate sustainable and user-centric design principles into product development processes. (*LO 4.1, LO 4.2, LO 6.1, LO 6.2*)

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
HSS301.1	-	3	3	-	-	3	-	-	-	-	-
HSS301.2	-	-	3	-	3	-	-	-	-	-	-
HSS301.3	-	-	--	-	-	-	-	3	-	-	-
HSS301.4	-	-	3	-	3	-	-	-	-	3	-
HSS301.5	-	-	3	-	-	3	3	-	-	-	3
Average	-	**3**	**3**	-	**3**	**3**	**3**	**3**	-	**3**	**3**

Text Books :

1. "Product Design and Development" by Karl T. Ulrich and Steven D. Eppinger, published by McGraw-Hill Education; 7th edition (January 25, 2021).
2. "Engineering Design: A Project-Based Introduction" by Clive L. Dym and Patrick Little, published by Wiley; 4th edition (August 26, 2015).
3. "Universal Principles of Design" by William Lidwell, Kritina Holden, and Jill Butler, published by Rockport Publishers; Revised and updated edition (January 1, 2010).

Reference Books :

1. "Sketching: Drawing Techniques for Product Designers" by Koos Eissen and Roselien Steur, published by BIS Publishers; 2nd edition (March 1, 2011).
2. "Materials and Design: The Art and Science of Material Selection in Product Design" by Mike Ashby and Kara Johnson, published by Butterworth-Heinemann; 3rd edition (October 10, 2014).
3. "The Design of Everyday Things" by Don Norman, published by Basic Books; Revisedand expanded edition (November 5, 2013).

Other Resources :

1. NPTEL Course: Product Design and Development, Prof. Inderdeep Singh, IITRoorkee Weblink: https://onlinecourses.nptel.ac.in/noc21_me83/preview
2. NPTEL Course: Product Design and Innovation, By Prof. Supradip Das, Prof. Swati Pal, Prof. Debayan Dhar, IIT Guwahati, IIT Guwahati, Web link: https://onlinecourses.nptel.ac.in/noc21 de01/preview

Continuous Assessment (50 Marks)

Suggested breakup of distribution

Multiple Choice Questions	10 Marks
Case Study	20 Marks
Group Project	15 Marks
Regularity and Active participation	05 Marks

Course Type	Course Code	Course Name	Credits
PCC	CSPCC405	ENGINEERING MATHEMATICS-IV	04

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Exam (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite :

1. BSC204 Engineering Mathematics-II
2. CSPCC301 Engineering Mathematics -III

Program Outcomes addressed :

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO5: Engineering tool usage
4. PO8: Individual and team work
5. PO11: Life Long Learning

Course Objectives :

1. To provide the basic knowledge on the concepts of Mathematics in the field of Engineering.
2. To build a foundation to the methodology necessary for solving problems by applying the knowledge of Mathematics to the field of Engineering.

Module	Details	Hrs
00.	**Course Introduction** Engineering Mathematics IV aims to equip students with the foundational knowledge and skills necessary for analyzing uncertainty and making informed decisions in engineering contexts. This course has many applications in Information Technology For example 1) Application of in Probability in Machine Learning, NLP part of speech Hidden Markov Model. 2) Application of in Correlation and Regression in Deep learning. 3) Application of Statistics in hypothetical or inferential learning. 4) Application of Algebraic Structure in Cyber Security and Cryptography. **MATLAB Introduction** As the course aims to bridge the gap between conceptual understanding and computational practice, MATLAB is introduced as a supporting tool. Fundamental operations and basic usage of MATLAB required for the course are taught prior to the formal commencement of the curriculum	**01** **01**

01.	**Probability Theory and Random Variable** ***Learning Objective/s:*** • *Analysis of random variables using the basic theory of probability.* • *Application of appropriate mathematical techniques to determine events involving conditional probability.*	**07-09**
	Contents: Conditional Probability, Bayes Theorem, Total Probability Theorem, Definition of Random Variable. Types of Random Variable: Discrete and Continuous, Probability Mass and Density Function. Measures of Central Tendency and Dispersion	
	Self-Learning Topics: *Cumulative Distribution and Density Function.*	
	Learning Outcomes: *A learner will be able to* *LO 1.1 Identify independents sets and disjoint sets and use its knowledge in the context of conditional probability. (P.I.-2.1.3)* *LO 1.2 Apply mathematical techniques of union, intersection and addition of sets, numbers for finding probabilities of events using Bayes' Theorem and Total Probability Theorem and thereby develop proficiency for reasoning under uncertainty. (P.I.-1.1.2, P.I.-11.1.1)* *LO 1.3 Identify if a given Random variable is Discrete or continuous in nature using existing definitions and formulas from Probability. (P.I.-2.1.1)* *LO 1.4 Apply mathematical techniques of integration and summation for finding Expectation, Variance, of the Random Variable. (P.I.-1.1.1)* *LO 1.5 Work together as a team and using MATLAB analyze the previous semester class average and variance for Engineering Mathematic-III and make conclusions about last semester overall class performance. (P.I.-5.2.2, P.I.-8.2.1, P.I.-11.3.2)*	
02.	**Probability Distribution** ***Learning Objective/s:*** • *Analysis and identification of standard probability distribution functions.* • *Application of the properties of these distributions to determine probabilities of various events.*	**08-10**
	Contents: Binomial distribution, Poisson Distribution , Fitting of a Poisson Curve, Gaussian Distribution, Normal Distribution (Standard Normal distribution , Reverse problem of Normal distribution)	
	Self-Learning Topics: *Joint Probability Distribution*	

	Learning Outcomes: *A learner will be able to* *LO 2.1 Apply mathematical techniques of exponents, algebra and basic probability for finding the probabilities of various events using Binomial, Poisson and Normal Distribution. (P.I.-1.1.1)* *LO 2.2 Apply the advance mathematical techniques of probability to find the probabilities of various standard probability distribution (P.I. -1.1.2)* *LO 2.3 Apply the knowledge of statistical tables in determining the probabilities associated with a standard normal curve. (P.I. 1.3.1)* *LO 2.4 Identify the area under a Standard Normal Curve (bounded or unbounded) and use its knowledge in the context of Normal Distribution. (P.I.-2.1.3)* *LO 2.5: Analyze the information present, in a given real world mathematical problem and apply the concept of Poisson or Binomial distribution to determine the solution thereby maintaining proficiency with probabilistic reasoning and statistical modeling. (P.I.-2.1.3, P.I.-11.1.1)* *LO 2.6: Identify whether Poisson distribution and/ or Binomial Distribution is applicable to a given problem using basic definitions of distribution (P.I.-2.2.3)* *LO 2.7: Work together as a team in analyzing the information present in a given case study and apply the concept of Normal Distribution to not only determine mathematical solution via MATLAB but also maintaining a conceptual and computational balance. (P.I.-5.2.2, P.I.-8.3.1, P.I.-11.3.1)*	
03.	**Sampling Theory-I** ***Learning Objective/s:*** • *Analysis of the principles and procedures of hypothesis testing* • *Application of appropriate parametric tests to draw conclusions regarding a hypothesis*	**08-10**
	Contents: Introduction to Sampling Theory, Testing of Hypothesis, level of significance, Critical region, One tailed and Two tailed test, Type-I and Type-II error, Students' t-distribution: single mean, difference between the two means. Large samples test: single mean, difference between the two means.	
	Self-Learning Topics: *Central Limit Theorem*	
	Learning Outcomes: ***A learner will be able to*** *LO 3.1 Apply the knowledge of statistical tables in determine the critical region and critical values for both small and large sample tests. (P.I. 1.3.1)* *LO 3.2 Apply basic mathematical knowledge of inequality, division, etc. in calculating the test statistic needed for testing the hypothesis. (P.I.-1.1.1)* *LO 3.3: Analyze the information present, in a given real world mathematical problem and apply a parametric test to determine the solution thereby maintaining proficiency to validate assumptions, evaluate performance and compare systems. (P.I.-2.1.3, P.I.-11.1.1)* *LO 3.4: Evaluate the given mathematical statements and based on the given information set the null and alternate hypothesis and support the hypothesis by testing it. (P.I.-2.1.1, P.I.-11.3.1)*	

04.	**Sampling Theory-II** ***Learning Objective/s:*** • *Analysis of the principles, assumptions, and procedures of hypothesis testing.* • *Application of suitable parametric or non-parametric tests to draw statistical conclusions.*	**07-09**
	Contents: Chi-square test: Test of goodness of fit, Independence of attributes (Contingency table), distribution of sample variance F-test, significant difference between variances of two Samples.	
	Self-Learning Topics: *Yate's Correction, ANOVA*	
	Learning Outcomes: *A learner will be able to* *LO 4.1: Apply the knowledge of statistical tables in determine the critical region and critical values for both Chi-square and F-test. (P.I. 1.3.1)* *LO 4.2: Evaluate the given mathematical statements and based on the given information set the null and alternate hypothesis for either a parametric or non-parametric test and support the hypothesis by testing it. (P.I.-2.1.1, P.I.-11.1.1)* *LO 4.3: Identity the appropriate Chi-square test which can be applied to test the given hypothesis using the basics of non-parametric tests. (P.I.-2.1.3)* *LO 4.4: Apply laws of natural science to infer data from the given mathematical problem and thereby, set and solve the hypothesis. (P.I.- 1.2.1)* *LO 4.5: Work together as a team and source information from case studies/research papers based on non-parametric test techniques and present the mathematical solution maintaining a conceptual and computational balance. (P.I.-8.3.1, P.I.-11.3.1)*	
05.	**Correlation and Regression** ***Learning Objective/s:*** • *Analysis of the given mathematical dataset to understand variable behavior* • *Application of correlation and regression techniques to identify relationships between variables*	**08-10**
	Contents: Correlation, Karl Pearson's coefficients of correlation(r), Spearman's Rank correlation coefficient (R): Repeated Rank, Non-repeated rank, Regression, Line of regression, Curve fitting: Linear and Second-Degree Curves.	
	Self-Learning Topics: *Fitting of an exponential Curve*	
	Learning Outcomes: *A learner will be able to* *LO 5.1 Identify whether Karl Pearson's or Spearman's coefficient of correlation is to be used in establishing relationship between two variables depending on the dataset given. (P.I.- 2.1.3)* *LO 5.2 Apply basic mathematical techniques from algebra in finding the lines of regression and regression coefficients. (P.I.-1.1.1)*	

	LO 5.3 Apply Least Square Method to fit a particular curve or line to the given data (P.I.-1.1.2) *LO 5.4 Identify whether a linear degree curve or a quadratic degree curve is to be fit for the given data set based on the knowledge of Curve Fitting (P.I.-2.2.3)* *LO 5.5 Work together as a team and predict and thereby present the End semester marks for students in the team using Regression and Curve Fitting. (P.I.- 5.1.1, P.I.- 8.3.1)*	
06.	**Algebraic Structure** ***Learning Objective/s:*** • *Analysis of various algebraic structures* • *Application of this knowledge to study and identify isomorphism between algebraic structures.*	**07-09**
	Contents: Groups: Definition and Examples, Rings, Integral domain, Fields, Ring Homomorphism, Ring Isomorphism	
	Self-Learning Topics: *Group Homomorphism and Group Isomorphism*	
	Learning Outcomes: *A learner will be able to* *LO 6.1 Apply mathematical operations defined on algebraic structures like Rings, Integral domain and Field and demonstrating closure properties under the operations. (P.I.-1.1.1)* *LO 6.2 Identify substructures within algebraic systems and the concept of homomorphism between them. (P.I.-2.1.3)* *LO 6.3 Apply the properties of homomorphism and one-one prove that the homomorphism is an isomorphism (P.I. 1.1.1)* *LO 6.4 Identify and characterize various algebraic structures based on their properties.(P.I.-2.4.1)*	
	Course Conclusion	**01**
	Engineering Mathematics plays an important role in providing the analytical tools necessary for designing, analyzing, and optimizing various electronic systems and communication networks.	
	Total	**60**

Performance Indicators:

P.I. No.	P.I. Statement
1.1.1	Apply the knowledge of discrete structures, linear algebra, statistics and numerical techniques to solve problems
1.2.1	Apply laws of natural science to an engineering problem
1.3.1	Apply engineering fundamentals
2.1.1	Evaluate problem statements and identifies objectives
2.1.3	Identify mathematical algorithmic knowledge that applies to a given problem
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions

5.1.1 Identify modern engineering tools, techniques and resources for engineering activities
5.2.2 Demonstrate proficiency in using discipline specific tools
8.2.1 Demonstrate effective communication, problem solving, conflict resolution and leadership skills
8.3.1 Present results as a team, with smooth integration of contributions from all individual efforts
11.1.1 Describe the rationale for requirement for continuing professional development
11.3.1 Source and comprehend technical literature and other credible sources of information
11.3.2 Analyse sourced technical and popular information for feasibility, viability, sustainability, etc.

Course Outcomes :

1. Identify the type random variables and apply the concepts of probability for getting the spread of data.*(LO 1.1, LO 1.2, LO 1.3 , LO 1.4)*
2. Analyse the given mathematical problem and apply the concepts of distribution in finding probabilities. *(LO 2.1, LO 2.2, LO 2.3 , LO 2.4, LO 2.5, LO 2.6)*
3. Identify whether the given problem is parametric or non-parametric and thereby apply various sampling theory principles and techniques to solve and interpret the result. *(LO 3.1, LO 3.2, LO 3.3 , LO 3.4, LO 4.1, LO 4.2, LO 4.3, LO 4.4)*
4. Analyse and interpret the data using Correlation and Regression. (*LO 5.1, LO 5.2, LO 5.3 , LO 5.4)*
5. To analyse various algebraic structures and apply it to determine isomorphism between any two structures. (*LO 6.1, LO 6.2, LO 6.3 , LO 6.4)*
6. Analyse and apply mathematical tools acquired through the course and thereby, exhibit the knowledge through written solutions or presentations as a team and/or individual activities. *(LO 1.5, LO 2.7, LO 4.5, LO 5.5)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC405.1	3	3	-	-	-	-	-	-	-	-	3
CSPCC405.2	3	3	-	-	-	-	-	-	-	-	3
CSPCC405.3	3	3	-	-	-	-	-	-	-	-	3
CSPCC405.4	3	3	-	-	-	-	-	-	-	-	-
CSPCC405.5	3	3	-	-	-	-	-	-	-	-	-
CSPCC405.6	-	-	-	-	3	-	-	3	-	-	3
Average	**3**	**3**	**-**	**-**	**3**	**-**	**-**	**3**	**-**	**-**	**3**

Text Books :

1. Probability, Statistics and Random Processes by T Veerarajan, Third Edition, Tata McGraw-Hill Publishing Company Limited.
2. Higher Engineering Mathematics by Grewal B. S. 38th edition, Khanna Publication 2005.

Reference Books :

1. Topics in Algebra by Herstein.
2. Fundamentals of Probability and Statistics V. K Kapoor and S. C. Gupta,. S . Chand, Publications
3. A Text Book of Applied Mathematics Vol. I & II by P.N.Wartikar & J.N.Wartikar, Pune, Vidyarthi Griha Prakashan, Pune.
4. MATLAB: An Introduction with Applications by Amos Gilat, Fourth edition, John Wiley and Sons(Wiley)

Other Resources :

1. NPTEL Course: NOC: Probability for Computer Science IIT Kanpur: Prof Nitin Saxena:-Web link- https://nptel.ac.in/courses/106104233/
2. NPTEL Course: Sampling Theory, IIT Kanpur: Prof Shalabh :-Web link https://nptel.ac.in/courses/111104073
3. NPTEL Course::- NOC: Algebra -I IMSc, Prof S Vishwanath, Prof Amritanshu Prasad Web link- https://nptel.ac.in/courses/111106137
4. NPTEL Course: MATLAB Programming for Numerical Computation: Prof. Niket Kaisare, Web link- https://nptel.ac.in/courses/103106118

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment - Theory (20 MARKS)

One MCQ test as per Gate exam pattern/ level: 5 Marks
One Class test: 5 Marks
One Team-pair- Solo: 5 Marks
Regularity and attentiveness: 5 Marks

2. Mid Semester Exam (30 Marks)

Mid semester examination will be based on 40% to 50% of the syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20%-30% weightage, and the syllabus covered from MSE to ESE carrying 70%-80% weightage.

Course Type	Course Code	Course Name	Credits
PCC	CSPCC406	DESIGN & ANALYSIS OF ALGORITHM	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Examination (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite :

1. CSPCC303: Data structure
2. CSPCC301: Engineering Mathematics-III

Program Outcomes addressed :

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3 : Design/development of solutions
4. PO 4: Conduct investigations of complex problems
5. PO8: Individual and Collaborative Team work
6. PO9: Communication
7. PO 11: Life–Long Learning

Course Objectives :

1. To provide mathematical background for analysis of algorithm
2. To introduce various advanced data structures.
3. To provide comprehensive knowledge about different techniques of problem solving/ algorithm design.
4. To make the students select and apply different algorithm design techniques for problem solving.
5. To familiarize learners with advanced algorithms and the concepts of pattern matching.

Module	Details	Hrs
00.	**Course Introduction** This course includes different techniques of problem solving and their applications for solving problems belonging to multidisciplinary domains. It also involves analysis of algorithms developed using these techniques, selection and application of an optimal technique/algorithm for solving a given problem. The course contents enable software engineers to make decisions on the design and implementation of more efficient algorithms, leading to faster and more scalable software systems in resource constrained diverse industry applications where high performance is a key factor.	01

01.	**Fundamentals of the analysis of algorithms and Advanced Data Structures** ***Learning Objective:*** *To introduce different methods to determine the time and space complexity of different algorithms and to select and apply the most suitable one for a particular problem.* *To introduce advanced data structures and operations on them.*	**06-08**
	Contents: Fundamentals of the analysis of algorithms: Time and Space complexity, Asymptotic analysis and notation, average and worst-case analysis, Recurrences: The substitution method, Recursive tree method, Masters method. **Advanced Data Structures:** Definition , properties and operations on B tree, B+ tree, Red-Black Trees. Topological Sort –Definition, DFS method, adjacency matrix method.	
	Self-Learning Topics: *Analysis of Time and space complexity of iterative and recursive algorithms.* *Implementation of B tree and Red-Black Tree.*	
	Learning Outcomes : *A learner will be able to* *LO 1.1: Apply different methods to determine topological ordering of a directed acyclic graph. (P.I.-1.3.1)* *LO 1.2: Apply search and insert operations on B/B+ Tree. (P.I.-1.4.1)* *LO 1.3: Compare B Tree and B+ tree and select appropriate data structure between the two for providing indexed sequential access of data. (P.I.-2.2.4)* *LO 1.4: Apply different methods to derive the time complexity of a given recursive algorithm. (P.I.-1.3.1)* *LO 1.5: Apply appropriate method to derive the time complexity of a given Iterative algorithm. (P.I.-1.4.1)* *LO 1.6: Analyze and interpret the time complexity of Iterative and recursive algorithms. (P.I.- 2.4.2)* *LO 1.7: Identify and Use appropriate method to determine the time complexity of a recurrence relation. (P.I.-2.2.3)* *LO 1.8: Identify and Apply appropriate method to determine the topological ordering of a given graph. (P.I.-2.2.3)*	
02.	**Divide and conquer technique and Greedy Algorithms** ***Learning Objective/s:*** *To make the learners master Divide and conquer and Greedy technique and make them apply these techniques to solve large scale complex and optimization problems.*	**07-09**
	Contents:	
	Divide and conquer technique: Introduction to Divide and conquer technique of problem solving. Advantages, disadvantages. Types of problems that can be solved using D&C. Binary Search, Merge sort and Quick sort, Finding minimum and maximum algorithm. Analysis of all these algorithms. **Greedy Algorithms :** Introduction to Greedy Algorithms. Advantages, disadvantages. Types of problems that can be solved using Greedy Algorithms. Knapsack	

	Problem, Job sequencing using deadlines, Optimal storage on tape, Optimal Merge Pattern. Analysis of all these algorithms.	
	Self-Learning Topics: *Implementation of minimum and maximum algorithm, Knapsack problem, Job sequencing using deadlines.*	
	Learning Outcomes : *A learner will be able to* *LO 2.1: Apply Greedy technique to solve a given problem. (P.I.-1.3.1)* *LO 2.2: Apply Divide and Conquer strategy to design an algorithm to solve a given problem (P.I.-1.4.1)* *LO 2.3: Identify and Use appropriate algorithmic design technique to solve a given problem. (P.I.-2.2.3)* *LO 2.4: Analyze the time complexity of algorithms designed using Divide and Conquer technique. (P.I.- 2.4.2)* *LO 2.5: Explore different algorithmic design alternatives for finding a minimum and maximum number in an array.(P.I.-3.2.1)* *LO 2.6: Produce different design solutions to search for a given number in an array. (P.I.-3.2.2*	
03.	**Dynamic Programming** ***Learning Objective:*** *To acquaint the knowledge of dynamic programming technique to solve and analyze optimization problems in diverse application domains.*	**06-08**
	Contents: **Dynamic Programming :** Introduction to Dynamic Programming. Advantages, disadvantages. Types of problems that can be solved using Dynamic Programming. Comparison of Dynamic Programming with D&C and Greedy algorithms. All pair shortest path, 0/1 knapsack, travelling salesman problem, Matrix Chain Multiplication, Optimal binary search tree. Analysis of All these algorithms.	
	Self-Learning Topics: *Implementation of All pair shortest path, 0/1 Knapsack and OBST..*	
	Learning Outcomes : *A learner will be able to* *LO 3.1: Apply dynamic programming to determine an optimal order of Multiplication for multiplying a given chain of matrices (P.I.-1.3.1)* *LO 3.2: Apply dynamic programming approach to solve a given problem. (P.I.-1.4.1)* *LO 3.3: Compare different methods of solving knapsack problem. Select and apply best technique to solve 0/1 knapsack problem (P.I.-2.2.4)* *LO 3.4: Identify and apply an appropriate technique to solve travelling salesman problem. (P.I.-2.2.3)* *LO 3.5: Explore different design alternatives for solving knapsack problem. (P.I.-3.2.1)* *LO 3.6: Produce different design solutions to find an optimal solution of a given problem. (P.I.-3.2.2)*	

	LO 3.7: Conduct investigation of complex problems to choose appropriate algorithm for their implementation. (P.I.-4.1.2) *LO 3.8: Use appropriate tool/technique to analyze the time complexity of given problem. (P.I.-4.3.1)*	
04.	**Backtracking and Branch and bound** ***Learning Objective:*** *To acquaint the learners about Backtracking and Branch and bound algorithmic design techniques and make them apply and analyze these techniques to solve a wide range of computational problems including optimization, constraint satisfaction, puzzle solving, graph algorithms etc.*	**06-08**
	Contents: **Backtracking** : Introduction to Backtracking. Types of (decision) problems that can be solved using Backtracking Algorithms. N-queen problem, Sum of subsets or Graph coloring problem. **Branch and Bound :** Introduction to Branch and Bound. Types of (optimization) problems that can be solved using Branch and Bound, Comparison between Backtracking and Branch and Bound. Travelling Salesperson Problem or 15 Puzzle problem. Analysis of All these algorithms.	
	Self-Learning Topics: *Implementation of N-queen problem, 15 Puzzle problem.*	
	Learning Outcomes : *A learner will be able to* *LO 4.1: Apply backtracking to solve a given problem (P.I.-1.3.1)* *LO 4.2: Apply Branch and Bound technique to solve a given problem. (P.I.-1.4.1)* *LO 4.3: Identify different algorithmic design techniques to solve a given problem. (P.I.-2.1.2)* *LO 4.4: Compare Backtracking and Branch and Bound technique and select the best technique for solving a given optimization problem (P.I.- 2.2.4)* *LO 4.5: Explore different design alternatives for solving 15 puzzle problem. (P.I.-3.2.1)* *LO 4.6: Produce different design solutions to find an optimal solution of a given problem. (P.I.-3.2.2)*	
05.	**String matching algorithms** ***Learning Objective/s:*** *To acquaint the learners about the knowledge of string matching algorithms and to make the learners apply and analyze string matching algorithms for identifying patterns in textual data.*	**07-09**
	Contents: Introduction to string matching algorithms. The naïve string-matching algorithm, Rabin Karp algorithm, Knuth Morris-Pratt algorithm, Longest Common Subsequence (LCS). Analysis of All these algorithms.	

	Self-Learning Topics: *Implementation of Robin Karp algorithm, KMP algorithm and LCS.*	
	Learning Outcomes : *A learner will be able to* *LO 5.1: Apply Knuth Morris-Pratt algorithm for pattern matching. (P.I.-1.3.1)* *LO 5.2: Apply pattern matching algorithm to compute the longest common subsequence for a given set of strings. (P.I.-1.4.1)* *LO 5.3: Apply Rabin Karp algorithm to find the given pattern in a given string. (P.I.- 2.4.1)* *LO 5.4: Analyze the time complexity of Naïve Pattern matching algorithm. (P.I.-2.4.2)*	
06.	**Introduction to Advanced Algorithms and NP problems** ***Learning Objective/s:*** *To introduce/identify advance algorithms that are used for solving compute intensive and optimization problems for some of which the exact algorithms are of very low efficiency and the properties of NP hard and NP Complete problems.*	**05-07**
	Contents: **Optimization Algorithms:** Genetic algorithm (GA), **Approximation Algorithms:** Vertex-cover problem, **Parallel Computing Algorithms:** Fast Fourier Transform, Introduction to P, NP, NP-Hard and NP-Complete Problems.	
	Self-Learning Topics: *Implementation of Genetic algorithm and Vertex-cover problem.* ***Learning Outcomes :*** *A learner will be able to* *LO 6.1: Apply parallel algorithm to compute Fast Fourier Transform. (P.I.-1.3.1)* *LO 6.2: Apply approximate algorithm to solve vertex cover problem. (P.I.-1.4.1)* *LO 6.3: Use genetic algorithm approach to solve an optimization problem. (P.I.-2.3.1)* *LO 6.4: Identify the limitations of P, NP, NP- Hard and NP- Complete problems. (P.I.- 2.4.3)* *LO 6.5: Identify new algorithmic trends or tools relevant to computational problem solving and report the results effectively in a team. (P.I.-2.2.3, P.I.-8.3.1, P.I.-9.3.1, P.I.- 11.2.1)* *LO 6.6: Apply newly learned algorithmic concepts for solving problems and produce clear well- constructed document of the solution effectively in a team. (P.I.-1.4.1, P.I.-8.2.1, P.I.-9.1.2, P.I.- 11.3.3)*	
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

P.I. No.P.I. Statement

1.3.1 Apply engineering fundamentals.

1.4.1 Apply theory and principles of computer science engineering to solve an engineering problem.

2.1.2 Identifies processes/modules/algorithms of a computer based system and parameters to solve a problem

2.2.3 Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions

2.2.4 Compare and contrast alternative solution/methods to select the best methods

2.3.1 Able to apply computer engineering principles to formulate modules of a system with required applicability and performance

2.4.1 Applies engineering mathematics to implement the solution.

2.4.2 Analyze and interpret the results using contemporary tools.

2.4.3 Identify the limitations of the solution and sources/causes.

3.2.1 Ability to explore design alternatives.

3.2.2 Ability to produce a variety of potential design solutions suited to meet functional requirements.

4.1.2 Ability to choose appropriate procedure/algorithm, data set and test cases.

4.3.1 Use appropriate procedures, tools and techniques to collect and analyse data

8.2.1 Demonstrate effective communication, problem solving, conflict resolution and leadership skills.

8.3.1 Present results as a team, with smooth integration of contributions from all individual efforts

9.1.2 Produce clear, well-constructed, and well-supported written engineering documents

9.3.1 Create engineering-standard figures, reports and drawings to complement writing and presentations

11.2.1 Identify historic points of technological advance in engineering that required practitioners to seek education in order to stay current

11.3.3 Apply the knowledge of technological advancements in engineering in order to stay current.

Course Outcomes : Learner will be able to

1. Apply mathematical aspects and fundamentals of AOA for analysis of algorithms. (LO1.4, LO1.5, LO1.7, LO 6.4)
2. Apply different operations on advanced data structures. (LO 1.1, LO1.2, LO1.3, LO1.8)
3. Select and apply appropriate algorithm design technique for solving the given problem. (LO 2.1, LO 2.2, LO 2.3, LO 2.5, LO 2.6, LO 3.1, LO 3.2, LO 3.3, LO 3.4, LO 3.5, LO 3.6, LO 3.7, LO 4.1, LO 4.2, LO 4.3, LO 4.4 , LO 4.5, LO 4.6, LO 5.1, LO 5.2, LO 5.3, LO 6.1, LO 6.2, LO 6.3)
4. Analyse complexity of different algorithms. (LO 1.6, LO 2.4, LO 5.4, LO 3.8)
5. Solve a given problem by selecting and applying an appropriate algorithmic design technique and present the solution effectively in a team. (LO 6.5, LO 6.6)

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC406.1	3	3	-	-	-	-	-	-	-	-	-
CSPCC406.2	3	3	-	-	-	-	-	-	-	-	-
CSPCC406.3	3	3	3	3	-	-	-	-	-	-	-
CSPCC406.4	3	3	-	-	-	-	-			-	
CSPCC406.5	3	3	-	-	-	-	-	3	3	-	3
Average	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**3**	**3**	**-**	**3**

Text Books :

1. Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein, 3rd Edition, 2009, MIT Press.
2. Algorithms: Design and Analysis, Harsh Bhasin, 2015, OXFORD University Press.
3. Fundamentals of Computer Algorithms, Horowitz, Sahani, Rajsekaran, Universities Press.
4. C and Data structures, Deshpande, Kakde, 1st Edition, 2003, Dreamtech Press.

Reference Books :

1. Data Structures and Algorithms in C++, Goodritch, Tamassia, Mount, WILEY India Edition, 2015.
2. Data Structures using C, Reema Thareja , Second Edition, 2014, OXFORD.
3. Data Structures and Algorithm Analysis in C, Mark A. Weiss, Second Edition, 2020, Pearson.
4. Optimization Algorithms and Applications, By Rajesh Kumar Arora, First Editon, 2015, Chapman and Hall.

Other Resources :

1. NPTEL Course: Design and Analysis of Algorithms By Prof. Madhavan Mukund, Chennai Mathematical Institute. Web link- https://nptel.ac.in/courses/106/106/106106131/
2. NPTEL Course: Image Processing using Python By Prof. Sourav Mukhopadhyay; Department of Computer Science and Engineering, IIT Kharagpur
   Web link- https://archive.nptel.ac.in/courses/106/105/106105164/
3. NPTEL Course: Computer Vision and Image Processing - Fundamentals and Applications By Prof. Tim Roughgarden, Department of Computer Science, Stanford University, Stanford.
   Web link- https://www.coursera.org/specializations/algorithms

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (20 Marks)

Suggested breakup of distribution

a) One MCQ test as per GATE exam pattern/level: 05 Marks
b) Class Test : 05 Marks
c) Think-pair-share worksheets: 05 Marks
d) Regularity and active participation in class :05 Marks

2. Mid Semester Exam (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B. END SEMESTER EXAM (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20% to 30% weightage, and the syllabus covered from MSE to ESE carrying 70% to 80% weightage.

Course Type	Course Code	Course Name	Credits
PCC	CSPCC407	OPERATING SYSTEM	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Exam (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite:

1. ESL103: Programming Laboratory-I (C)
2. CSPCC303: Data Structures and Analysis

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of Solutions
4. PO4: Conduct Investigations of Complex Problems
5. PO5: Engineering Tool Usage
6. PO11: Life-Long Learning

Course Objectives:

1. Interpret the fundamental concepts, structure, and functions of operating systems.
2. Explore the principles of process management, including scheduling, synchronization, and inter-process communication.
3. Depict memory management strategies like as paging, segmentation, and virtual memory
4. Explicate secondary storage structure and file system management techniques.
5. Review the operating system principles of deadlock handling, security, and protection.

Module	Details	Hrs
00.	**Course Introduction** This course consists of concept of operating system, basic information of computer, Purpose of computer and its various components, Memory hierarchy. This course is used to understand the interaction between various hardware's and software applications.	01
01.	**Introduction of Operating System** ***Learning Objective:*** *To know and understand OS and its functions to solve real time-complex issues and maximize computational speed.*	06-08
	Contents: Introduction: Types of Computer System, Computer architecture, OS Structure, Operations, Services Interface, System Calls, System Structure, System Design and Implementation, OS Kernel, Types of kernels, Types of special purpose OS.	

	Self-Learning Topics: *Various types of computer and network system architectures.* ***Learning Outcomes:*** *A learner will be able to* *LO 1.1: Apply the concept of computer engineering fundamentals such as queuing theory in modeling of computer-based system and applications. (P.I. 1.1.2)* *LO 1.2: Apply kernel function theory and concepts to comprehend operating systems. (P.I.-1.4.1)* *LO 1.3: Identify appropriate system call for solving specific kernel function. (P.I.-2.1.2)* *LO 1.4: Identifies functionalities of OS and computing resources associated with OS kernel. (P.I.- 2.2.2)* *LO 1.5: Explore design structures of various system architecture w.r.t. need of OS (P.I.-3.2.1)* *LO 1.6: Choose appropriate OS based on need of operations by users. (P.I.- 3.2.2.)*	
02.	**Process Management** ***Learning Objective/s:*** *To know process management and apply process and thread scheduling.* **Contents:** Process Management: Process, Scheduling Inter-process Communication, Thread and its execution, CPU Scheduling: CPU Schedulers, scheduling Criteria Scheduling Algorithms. ***Self-Learning Topics:*** *Thread scheduling and system thread functions.* ***Learning Outcomes:*** *A learner will be able to* *LO 2.1: Apply computer engineering fundamentals to use process scheduling, IPC and thread scheduling. (P.I. 1.3.1)* *LO 2.1: Apply CPU scheduling techniques to ensure system operations are completed on time. (P.I.-1.4.1)* *LO 2.2: Identify appropriate scheduling method for task scheduling effectively. (P.I.-2.1.2)* *LO 2.3: Identify the limitations of selected process scheduling algorithm. (P.I.- 2.4.3)* *LO 2.4: Define a various function for thread scheduling over need of main process. (P.I.- 3.1.1)* *LO 2.5: Explore design of kernels w.r.t. OS functionalities. (P.I.-3.2.1)* *LO 2.6: Validate the kernel design based on scheduling techniques and their functions and choose appropriate solution. (P.I.-4.1.2)* *LO 2.7: Design and develop appropriate procedures/methodologies based on the scheduling policies types. (P.I.-4.2.1)*	**07-09**

03.	**Process Coordination**	**09-11**
	Learning Objective: *To know the process coordination and deadlock management policies in OS.*	
	Contents:	
	Process Synchronization: Critical Section, remainder section, Process Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Deadlocks: Characterization Methods for Handling Deadlocks, Deadlock Prevention, Avoidance, Detection, Recovery.	
	Self-Learning Topics: *Study a real time case study for Deadlock detection and recovery.*	
	Learning Outcomes: *A learner will be able to* *LO 3.1: Apply computer engineering fundamentals to know various fundamental in process structures, hardware synchronization and process critical section problems. (P.I. 1.3.1)* *LO 3.2: Apply several inter-process coordination algorithms and summarize the requirements. (P.I.- 1.4.1)* *LO 3.3: Identify appropriate semaphore and model system functions. (P.I.-2.1.2)* *LO 3.4: Identify the limitations of selected semaphore w.r.t. defined algorithm. (P.I.- 2.4.3)* *LO 3.5: Define deadlock in system and its causes. (P.I.- 3.1.1)* *LO 3.6: Verify the various process orderings for deadlock avoidance and detections based on obtained result. (P.I.- 3.4.3)* *LO 3.7: Able to choose appropriate procedure/algorithm on given process to avoid deadlock in the system. (P.I.-4.1.2)* *LO 3.8: Design and develop appropriate procedures for efficient interprocess communication. (P.I.-4.2.1)*	
04.	**Memory Management**	**06-08**
	Learning Objective: *To primary memory management policies and analyse the various page scheduling algorithms*	
	Contents: Memory Management: Hardware Address, Binding Address, Space Dynamic Loading and Linking, Swapping, Contiguous Allocation, Segmentation, Paging Structure of the Page Table, TLB, Virtual Memory Management: Demand Paging Page Replacement Algorithms, Thrashing and multiprogramming.	
	Self-Learning Topics: *Memory management for any one Operating System, Implementation of Page Replacement Algorithms.*	
	Learning Outcomes: *A learner will be able to*	

	LO 4.1: Apply the concepts of computer-based system to explore memory management unit of operating system. (P.I. 1.2.1) *LO 4.2: Apply several page scheduling strategies to a given collection of pages and assess the results. (P.I.- 1.4.1)* *LO 4.3: Identifies functionalities of various page scheduling algorithms. (P.I.- 2.2.2)* *LO 4.4: Identify appropriate page scheduling method on given pages of memory. (P.I.-2.1.2)* *LO 4.5: Compare and contrast between memory paging and segmentation techniques. (P.I.- 2.2.4)* *LO 4.6: Define virtual memory and explore it based on types of kernels. (P.I.- 3.1.1)* *LO 4.7: Explore design alternatives in case of virtual memory management. (P.I. 3.2.1)* *LO 4.7: Choose appropriate paging procedure/algorithm based upon obtained results. (P.I.-4.1.2)* *LO 4.8: Design and develop appropriate procedures for efficient paging/segmentation. (P.I.-4.2.1)*	
05.	**Disk and I/O Management** ***Learning Objective:*** *To know and understand File and directory and I/O management and compare various disk scheduling policies.*	**06-08**
	Contents: File system: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: Structure; implementation; Directory implementation; Allocation methods; RAID levels. I/O Systems: Overview I/O Hardware Application, I/O Interface, Kernel I/O, Subsystem Transforming, 1/0 Requests to Hardware Operations Performance. System Protection: Goals, Domain Access matrix, System Security: The Security Problem, Threats Encryption, User Authentication.	
	Self-Learning Topics: *File System for Linux and Windows, Features of I/O facility for different OS.*	
	Learning Outcomes: *A learner will be able to* *LO 5.1: Apply the concepts of computer-based system to explore file systems and I/O management. (P.I. 1.2.1)* *LO 5.2: Apply several disk scheduling strategies to a given collection of track/sectors of disks and assess the results. (P.I.- 1.4.1)* *LO 5.3: Identifies functionalities of various disk structures and their allocation, by applying engineering fundamentals. (P.I.- 2.2.2)* *LO 5.4: Identify appropriate disk scheduling method on given disk track/sectors sequences. (P.I.-2.1.2)*	

	LO 5.5: Compare and contrast between various RAID levels for disk allocation. (P.I.-2.2.4) *LO 5.6: Identify the limitations of the disk scheduling solution on given set of disk tracks and sectors. (P.I.-2.4.3)* *LO 5.7: Explore design alternatives and produce variety of design solutions. (P.I. 3.2.1, P.I. 3,.2.2)* *LO 5.8: Choose appropriate disk scheduling procedure/algorithm based upon obtained results. (P.I.-4.1.2)* *LO 5.9: Design and develop appropriate procedures for file system management and I/O frameworks. (P.I.-4.2.1)*	
06.	**Virtual Machines**	**03-05**
	Learning Objective: *To know and understand the demonstration of virtual machine on host and guest OS.*	
	Contents: Type 0, 1, and 2 hypervisors, Advantages of virtual machines, Trap and emulate method, Handling system calls, interrupts, and privileged instructions in VMs, Binary translation and paravirtualization, Type 2 Hypervisors, Shadow and nested page tables, JVMs and application-level hypervisors, Resource allocation in virtual machines, VM migration, Storage virtualization.	
	Self-Learning Topics: *Study and use of cloud based virtual machines.*	
	Learning Outcomes:	
	A learner will be able to *LO 6.1: Apply the concepts of computer-based system to explore virtualization and its integration in distributed environment. (P.I. 1.2.1)* *LO 6.2: Identifies functionalities and computing resources of virtual OS by applying engineering fundamentals. (P.I.-2.2.2)* *LO 6.3: Identifies processes/modules/algorithms of a computer-based system and parameters to solve a problem. (P.I.-2.1.2)* *LO 6.4: Refine guest architecture design into a detailed design within the existing operating system requirements and integrate various modules. (P.I. 3.4.2)* *LO 6.5: 3.4.3 Verify the functionalities and validate the design alternatives in case of virtual environment (P.I. 3.4.3)* *LO 6.6: Choose appropriate hardware/software tools to use Virtual OS on host OS. (P.I.-4.1.3)* *LO 6.7: Use appropriate procedures, tools and techniques to collect and analyze functionalities of VM. (P.I.: -4.3.1)* *LO 6.8: Identify modern engineering tools and discuss limitations of resources/ methods to enhance functionalities of Virtual OS on existing operating systems (P.I.-5.3.1)* *LO 6.9: Identify the strengths and limitations of INPUT/OUTPUT tools for (i) acquiring information, (ii) modeling and simulating, (iii) monitoring system*	

	performance, and (iv) develop emerging technologies solutions. (P.I. 5.2.1, P.I. 11.2.1) *LO 6.10: Recognize cost effectiveness of VM and usage with respect to new developments in emerging field by applying OS principles. (P.I. 1.4.1., P.I.-11.2.2)*	
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem.
2.1.2	Identifies processes/modules/algorithms of a computer.
2.2.2	Identifies functionalities and computing resources.
2.4.3	Identify the limitations of the solution and sources/causes.
3.1.1	Able to define a precise problem statement with objectives and scope.
3.2.1	Ability to explore design alternatives.
3.4.1	Ability to refine architecture design into a detailed design within the existing constraints.
3.4.3	Ability to verify the functionalities and validate the design.
4.1.2	Ability to choose appropriate procedure/algorithm, data set and test cases.
4.1.3	Ability to choose appropriate hardware/software tools to conduct the experiment.
4.3.1	Use appropriate procedures, tools and techniques to collect and analyze data.
5.1.1	Identify modern engineering tools, techniques and resources for engineering activities
5.2.1	Identify the strengths and limitations of tools for (i) acquiring information, (ii) modelling and simulating, (iii) monitoring system performance, and (iv) creating engineering designs.
11.2.1	Identify historic points of technological advance in engineering that required practitioners to seek education in order to stay current.
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field.

Course Outcomes:

Learner will be able to

1. Explore the fundamental concepts, structure, and functions of operating systems. *(LO1.1,LO 1.2, LO1.3, LO 1.4, LO 1.5)*
2. Analyze process management techniques including scheduling, synchronization, and deadlock handling. *(LO2.1..LO 2.2, LO2.3, LO 2.7, LO3.1, LO 3.2, ..LO3.8)*
3. Apply memory management strategies such as paging, segmentation, and virtual memory in system design. *(LO4.1, LO 4.2, LO4.3, LO 4.4, LO 4.5, LO4.6, LO 4.7, LO 4.8)*
4. Apply disk management strategies such as organization, scheduling, access methods of file systems and secondary storage. *(LO5.1, LO 5.2, LO5.3, LO 5.4.. LO 5.8)*
5. Examine the concepts of protection and security in operating systems and virtual machine to ensure safe system operation. *(LO6.1, LO 6.2, LO6.3, LO 6.4, LO 6.5.. LO 6.8)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC407.1	3	3	3	-	-	-	-	-	-	-	-
CSPCC407.2	3	3	3	3	-	-	-	-	-	-	-
CSPCC407.3	3	3	3	3	-	-	-	-	-	-	-
CSPCC407.4	3	3	3	3	-	-	-	-	-	-	-
CSPCC407.5	3	3	3	3	3	-	-	-	-	-	3
Average	**3**	**3**	**3**	**3**	**3**	-	-	-	-	-	**3**

Text Books:

1. Operating System Concepts, Abraham Silberschatz, Peter B Galvin, Greg Gagne, 9th Edition, Wiley India Pvt. Ltd 2018.
2. Operating Systems: Principles and Practice, Thomas Anderson and Michael Dahlin, 2nd Edition, 2014, Recursive Books.

Reference Books:

1. Operating Systems Internals and Design Principles, William Stallings, 9th Edition, 2018, Pearson.
2. Modern Operating Systems, Andrew S. Tanenbaum, Herbert Bos, 4th Edition, 2014, Pearson.

Other Resources:

1. Digital material:
   NPTEL: Introduction to Operating Systems by Prof. Chester Rebiero, IIT Madras
   Web Link-https://nptel.ac.in/courses/106106144
2. Digital material:
   NPTEL: Operating System Fundamental by Prof.Santanu Chattopadhay,IIT Kharagpur
   Web Link-https://archieve.nptel.ac.in/courses/106/105/106105214

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (20 Marks)

Suggested breakup of distribution

a) One MCQ Test as per GATE exam pattern / level: 05 Marks

b) One Class Test: 05 Marks

c) Mind Map activity: 05 Marks

d) Regularity and active participation :05 Marks

2. Mid Semester Examination (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20% to 30% weightage, and the syllabus covered from MSE to ESE carrying 70% to 80% weightage

Course Type	Course Code	Course Name	Credits
PCC	CSPCC408	SOFTWARE ENGINEERING	03

Examination Scheme					
Distribution of Marks			Exam Duration (Hrs.)		Total Marks
In-semester Assessment		End Semester Exam (ESE)			
Continuous Assessment	Mid-Semester Exam (MSE)		MSE	ESE	
20	30	50	1.5	2	100

Pre-requisite: NIL

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Development of Solutions
4. PO4: Conduct investigations of complex problems
5. PO5: Engineering tool usage
6. PO9: Communication
7. PO11: Life-long learning

Course Objectives :

1. To familiarize learner with the basic knowledge in software engineering.
2. To acquaint leaners to identify requirements, analyse and prepare models.
3. To familiarize learner to plan, schedule and track the progress of the projects.
4. To introduce leaners to design & develop the software solutions for the growth of society.
5. To familiarize learner to demonstrate and evaluate real time projects with respect to software engineering principles.
6. To introduce leaners to apply testing and assure quality in software solution.

Module	Detailed Contents	Hrs
00.	**Course Introduction** Software Engineering Course deals with the basic principles of Software Process, with emphasis on the Process models and their usage for different types of Software. This is foundation course which deals with fundamental concepts of Requirement Analysis, Software Estimation and Scheduling, Risk and Configuration Management, Software Testing and Maintenance. The fundamental concepts of this subject are essential for working on Software Projects in Industry.	01
01.	**Introduction to Software Engineering** ***Learning Objective/s:*** *Learner is expected to know the nature of software, software process and the different process models.*	06-08

	Contents:	
	Nature of Software, Software Engineering, Software Process, Capability Maturity Model (CMM) Generic Process Model, Prescriptive Process Models: The Waterfall Model, V-model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Agile process, Agility Principles, Extreme Programming (XP), Scrum, Kanban model	
	Self-Learning Topics: *Personal and Team Process Models.*	
	Learning Outcomes : *A learner will be able to* *LO 1.1: Apply concepts of Software Engineering Principles to create report for documenting standard Software Engineering processes followed for development of a software. (P.I.-1.3.1) (P.I.-9.3.1)* *LO 1.2: Use core principles of Software Engineering to understand the importance of logical progression of software engineering practices in design and development of software applications. (P.I.-1.4.1) (P.I.-9.1.3)* *LO 1.3: Compare different Process models to select the best process model that suits a particular software. (P.I.-2.2.4)* *LO 1.4: Identify the level of software according to CMM model. (P.I.-2.3.1)*	
02.	**Requirement Analysis** ***Learning Objective/s:*** *Learner is expected to know and understand the concepts of requirement analysis in software engineering.*	**07-09**
	Contents: Software Requirements: Functional & non-functional – user-system requirement engineering process – feasibility studies – elicitation – validation & management – software prototyping – S/W documentation – Analysis and modeling Requirement Elicitation, Software requirement specification (SRS). Unified Modeling Language diagrams.	
	Self-Learning Topics: *Prioritizing requirements (Kano diagram) - real life application case study.*	
	Learning Outcomes : *A learner will be able to* *LO 2.1: Apply basic concepts of feasibility studies to check if a particular software is feasible for an organization. (P.I.-1.3.1)* *LO 2.2: Use requirement elicitation techniques to know about customer requirements. (P.I.-1.4.1)* *LO 2.3: Identify requirements and parameters to solve a software based problem (P.I.-2.1.2)* *LO2.4: Identify appropriate software prototyping model to solve a given problem. (P.I.-2.2.3)* *LO 2.5: Determine requirements of a given problem and develop Software Requirement Specifications-SRS for it. (P.I.-3.1.6)* *LO 2.6: Identify suitable functional requirements for producing a variety of design solutions using UML for software. (PI-3.2.2)*	
03.	**Software Estimation and Scheduling** ***Learning Objective/s:*** *Learner is expected to know and understand the concepts of software estimation and scheduling.*	**06-08**

	Contents:	
	Management Spectrum, 3Ps (people, product and process) Process and Project metrics Software Project Estimation: LOC, FP, Empirical Estimation Models - COCOMO II Model, Specialized Estimation Techniques, Object based estimation, use-case based estimation Project scheduling: Defining a Task Set for the Software Project, Timeline charts, Tracking the Schedule, Earned Value Analysis	
	Self-Learning Topics: *Typical Problems with IT cost estimates.* ***Learning Outcomes :*** *A learner will be able to* *LO 3.1: Apply engineering fundamentals to identify Process and Project metrics for a software. (P.I.-1.3.1)* *LO 3.2: Use empirical estimation models to estimate cost of software. (P.I.-1.4.1)* *LO 3.3: Estimate the cost and identify resources required for a software using Function point analysis. (P.I.-2.2.2)* *LO 3.4: Compare scheduling techniques and select best scheduling technique for a software. (P.I.-2.2.4)*	
04.	**Design Engineering** ***Learning Objective/s:*** *Learner is expected to know and understand concepts of design engineering.*	**06-08**
	Contents: Design Process & quality, Design Concepts, The design Model, Pattern-based Software Design. Architectural Design: Design Decisions, Views, Patterns, Application Architectures, Modeling Component level Design: component, Designing class based components, conducting component-level design.	
	Self-Learning Topics: *Refinement aspects, Refactoring*	
	Learning Outcomes : *A learner will be able to* *LO 4.1: Use principles of architectural design to design a given software based application. (P.I.-1.3.1)* *LO 4.2: Apply the concepts of Pattern based Software design to use in a software system. (P.I.-1.4.1)* *LO 4.3: Identify the architectural view that best suits a software. (P.I.-2.1.2)* *LO 4.4: Identify the class based components to be used in component level design. (P.I.-2.2.3)* *LO 4.5: Use different methods to design a given software based application P.I.-3.2.1)* *LO 4.6: Design architectural diagram/class based components to meet functional requirements of a given application. (P.I.-3.2.2)*	
05.	**Software Testing** ***Learning Objective/s:***	**08-10**

	Learner is expected to know and understand the concepts of software risk and configuration management.	
	Contents: Testing: Software Quality, Testing: Strategic Approach, Strategic Issues- Testing: Strategies for Conventional Software. Dynamic Testing: Black Box Testing: Boundary Value Analysis, Equivalence Class Partitioning, State Table Based testing, Cause-Effect Graphing Based Testing, Error Guessing. White Box Testing Techniques: need, Logic Coverage Criteria, Basis Path Testing, Graph Matrices, Loop Testing, Static Testing: Formal Technical Review (FTR), Walkthrough. Study of testing tools**.** Dynamic and static testing tools JIRA, Bugzilla.	
	Self-Learning Topics: *Configuration Management for Web Apps.*	
	Learning Outcomes : *A learner will be able to* *LO 5.1: Use different testing techniques to test a given software application. (P.I.-1.3.1)* *LO 5.2: Apply Boundary Value Analysis or Equivalence Class Partitioning to test a given software application. (P.I.-1.4.1)* *LO 5.3: Identify processes/algorithms to design test cases to test a given software application. (P.I.-2.1.2)* *LO 5.4: Compare different methods of testing a software application. Select and apply the best suitable testing technique for testing a given software application(P.I.-2.2.4)* *LO 5.5: Apply different techniques to design test cases to identify state of bugs. (P.I.-3.2.1)* *LO 5.6: Determine the functionalities and validate the test cases designed. (P.I.-3.4.3)* *LO 5.7: Choose appropriate procedure/algorithm, data set and test cases to find structural errors. (P.I.-4.1.2)* *LO 5.8: Analyze the structural bugs and logical bugs by running the test cases to refine the process over time. (P.I.-4.3.2)* *LO 5.9: Identify and Use new testing tools and techniques to design test data and test cases (P.I.-5.2.2) (P.I.-11.2.2).* *LO 5.10: Evaluate the credibility of results from tools used with reference to the accuracy and limitations. (P.I.-2.4.3, P.I.-5.3.2, P.I.-11.1.2).*	
06.	**Software Risk, Configuration Management and Software Quality Assurance** ***Learning Objective/s:*** *Learner is expected to know and understand concepts of software testing and assure quality of software solution.*	**04-06**
	Contents:	

	Risk Identification, Risk Assessment, Risk Projection, RMMM Software Configuration management, SCM repositories, SCM process. Software Quality Assurance Task and Plan, Metrics, Software Reliability, Software Quality Management and Assurance, McCall's quality factors and criteria, ISO9000:2000, SIX sigma. Basics of Software Maintenance and Reverse Engineering.	
	Self-Learning Topics: *Web based and mobile based Software quality assurance factors and criteria.* ***Learning Outcomes :*** *A learner will be able to* *LO 6.1: Use RMMM to identify, assess and project risks in projects. (P.I.-1.3.1)* *LO 6.2: Apply SCM Process to solve an engineering problem associated with software. (P.I.-1.4.1)* *LO 6.3: Identify quality attributes for software quality assurance (P.I.-1.3.1).* *LO 6.4: Apply principles of Software Quality Management and Assurance to identify quality factors and criteria for a given application. (P.I.-1.4.1)* *LO 6.6: Apply the concepts of Six Sigma, Software Maintenance or Reverse Engineering for a given task.* (P.I.-1.4.1) *LO 6.7: Analyze the impact of Six Sigma on software quality attributes.* (P.I.-2.4.2) *LO 6.8: Analyze and Apply McCall's criteria to for a given software system with required applicability and performance (P.I.-2.3.1)* *LO 6.9: Identify and Analyze quality attributes for required performance criteria. (P.I.-2.3.2)*	
	Course Conclusion	**01**
	Total	**45**

Performance Indicators:

P.I. No.	P.I. Statement
1.3.1	Apply engineering fundamentals
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem
2.1.2	Identifies processes/modules/algorithms of a computer based system and parameters to solve a problem
2.2.2	Identifies functionalities and computing resources.
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions
2.2.4	Compare and contrast alternative solution/methods to select the best methods
2.3.1	Able to apply computer engineering principles to formulate modules of a system with required applicability and performance
2.3.2	Identify design constraints for required performance criteria.
2.4.3	Identify the limitations of the solution and sources/causes.
3.1.6	Ability to develop software requirement specifications (SRS).

3.2.1	Ability to explore design alternatives.
3.2.2	Ability to produce a variety of potential design solutions suited to meet functional requirements.
3.4.2	Ability to implement and integrate the modules.
3.4.3	Ability to verify the functionalities and validate the design.
4.1.2	Ability to choose appropriate procedure/algorithm, data set and test cases.
4.3.2	Critically analyse data for trends and correlations, stating possible errors and limitations
5.2.2	Demonstrate proficiency in using discipline specific tools.
5.3.2	Verify the credibility of results from tool use with reference to the accuracy and limitations, and the assumptions inherent in their use.
9.1.3	Create flow in a document or presentation - a logical progression of ideas so that the main point is clear
9.3.1	Create engineering-standard figures, reports and drawings to complement writing and presentations
11.1.2	Identify deficiencies or gaps in knowledge and demonstrate an ability to source information to close this gap
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field

Course Outcomes:

1. Use of basic knowledge in software engineering. *(LO 1.1, LO1.2, LO1.3, LO1.4)*
2. Identify requirements, analyse and prepare models. *(LO 2.1, LO 2.2, LO 2.3, LO 2 .4, LO 2.5, LO 2.6)*
3. Plan, schedule and track the progress of the projects.(*LO 3.1, LO 3.2, LO 3.3, LO 3.4)*
4. Design & develop the software solutions for the growth of society. (*LO 4.1, LO 4.2, LO 4.3, LO 4.4, LO4.5, LO4.6)*
5. Apply testing and assure quality in software solution. *(LO 5.1, LO 5.2, LO 5.3, LO 5 .4, LO 5.5, LO 5.6, LO 5.7, LO5.8, LO5.9, LO5.10, LO 6.1, LO 6.2, LO 6.3, LO 6.4, LO 6.5, LO 6.6, LO 6.7, LO 6.8, LO 6.9)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSPCC408.1	3	3	-	-	-	-	-	-	3	-	-
CSPCC408.2	3	3	3	-	-	-	-	-	-	-	-
CSPCC408.3	3	3	-	-	-	-	-	-	-	-	-
CSPCC408.4	3	3	3	-	-	-	-	-	-	-	-
CSPCC408.5	3	3	3	3	3	-	-	-	-	-	3
Average	**3**	**3**	**3**	**3**	**3**	-	-	-	**3**	-	**3**

Text Books :

1. Software Engineering: A Practitioner‘s Approach, Roger Pressman, 9th edition , 2019, McGraw-Hill Publications
2. Software Engineering , Ian Sommerville, 9th edition, 2011, Pearson Education
3. Software Engineering Fundamentals , Ali Behfrooz and Fredeick J. Hudson, 9th edition, 1997, Oxford University Press.

4. The unified modeling language user guide, Grady Booch, James Rambaugh, Ivar Jacobson, Second Edition, 2005, Pearson Education

Reference Books :

1. An integrated approach to Software Engineering, Pankaj Jalote, Third edition, 2005, Springer publication.
2. Fundamentals of Software Engineering , Rajib Mall, Fifth Edition, 2014, Prentice Hall India
3. Software Engineering, Jibitesh Mishra and Ashok Mohanty, Third edition, 2011, Pearson Education
4. Software Engineering – Concepts and Practices, Ugrasen Suman, Third edition, 2013, Cengage Learning
5. Software Engineering principles and practice, Roger Pressman, Second edition , 2004, McGraw-Hill Publications

Other Resources:

1. NPTEL Course: Software Engineering By Prof. Rushikesh Joshi, Prof. Umesh Bellur, Prof. N.L. Sarda, Department of Computer Science and Engineering, IIT Bombay :-Web link- https://nptel.ac.in/courses/106/101/106101061/
2. NPTEL Course: Software Engineering By Prof. Rajib Mall, Department of Computer Science and Engineering, IIT Kharagpur :-Web link- https://nptel.ac.in/courses/106105087

A. IN-SEMESTER ASSESSMENT (50 MARKS)

1. Continuous Assessment (20 Marks)

Suggested breakup of distribution

a) One MCQ Test as per GATE exam pattern / level: 05 Marks

b) One Class test:05 Marks

c) One Mind Map activity: 05 Marks

d) Regularity and active participation :05 Marks

2. Mid Semester Examination (30 Marks)

Mid semester examination will be based on 40% to 50% syllabus.

B. END SEMESTER EXAMINATION (50 MARKS)

End Semester Examination will be based on syllabus coverage up to the Mid Semester Examination (MSE) carrying 20% to 30% weightage, and the syllabus covered from MSE to ESE carrying 70% to 80% weightage.

Course Type	Course Code	Course Name	Credits
LBC	CSLBC403	DESIGN AND ANALYSIS OF ALGORITHM LABORATORY	01

Examination Scheme		
Continuous Assessment	**End Semester Exam(ESE)**	**Total**
25	25	50

Pre-requisite:

1. ESL103: Programming Laboratory-I (C)
2. ESL205: Programming Laboratory-II (Java)
3. CSSBL301 : Python Laboratory

Program Outcomes addressed:

1. PO 1: Engineering knowledge
2. PO 2: Problem analysis
3. PO 3: Design/Development of Solutions
4. PO 4: Conduct investigations of complex problems
5. PO 5: Engineering tool usage
6. PO 9: Communication
7. PO 11: Life–Long Learning

Course Objectives:

1. To make the learners implement different approaches of algorithm design.
2. To make the learners determine the time complexity of algorithms.
3. To make learners implement different algorithmic design techniques for solving a given problem.

Module	Detailed Contents	Hrs
00.	**Course Introduction** This course includes designing and analysing efficient algorithms to solve computational problems belonging to multidisciplinary domains. It involves applying different algorithmic design techniques such as divide-and-conquer, greedy, dynamic programming, branch and bound, backtracking, string matching and graph algorithms to solve a given problem. The course contents enable software engineers to make informed decisions on the design and implementation of efficient algorithms for solving a given problem. This is essential for building high-performance, resource-efficient, and scalable software systems in diverse application domains where algorithmic efficiency plays a critical role.	
01.	**Fundamentals of the analysis of algorithms**	**08**

	Learning Objective/s: *The learner is expected to implement different approaches of algorithm design, determine the time complexity of algorithms designed using these approaches and select and apply the most suitable one for a particular problem.* **Contents:** Types of algorithms – Iterative, recursive algorithms, Time complexity. **Task 1:** Implement a given problem using iterative and recursive approach and analyze its time complexity using both approaches to select the approach that requires minimal time. ***Learning Outcome :*** *A learner will be able to* *LO 1.1: Explore different algorithmic design alternatives and apply them to design an algorithm to solve a given problem. Analyze and interpret the difference between alternative algorithmic design methods to select the most appropriate method based on time complexity. Evaluate systematically the time complexity of iterative and recursive approach to select optimal design scheme for further development.* *(P.I.-1.3.1)(P.I.-1.4.1) (P.I.-2.2.4)(P.I.-2.4.2) (P.I.-3.2.1) (P.I.-3.3.1)*	
02.	**Divide and conquer technique and Greedy Algorithms** ***Learning Objective/s:*** *The learner is expected to implement Divide and conquer and Greedy technique of problem solving and apply these techniques to solve large scale complex and optimization problems.* **Contents :** Divide and conquer technique of problem solving. Greedy technique of problem solving. **Task 2:** Design and implement appropriate algorithmic design technique to solve the problem given in 2a, 2b. Analyze the limitations of existing methods. Explore alternative design schemes. Evaluate which method is optimal for a given application/problem statement and Use appropriate tools/technique (e.g., Divide & Conquer, Greedy) to implement the solution. 2a. Search for a given number in an array using binary search. 2b. Determine Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm. ***Learning Outcome:*** *A learner will be able to* *LO 2.1: Analyze a given problem to identify existing methods to solve it. Evaluate its requirements systematically, explore different algorithmic design alternatives and apply appropriate tools/techniques such as divide & conquer, greedy approaches to solve it. Identify the limitations of both the techniques and select an optimal design scheme for further development. (P.I.-1.3.1) (P.I.-1.4.1) (P.I. - 2.2.3) (P.I.-2.4.3) (P.I.-3.2.1)(P.I.-3.3.1)(P.I.- 5.1.2)(P.I.-5.3.1)*	**04**
03.	**Dynamic Programming** ***Learning Objective/s:*** *Learner is expected to apply the knowledge of dynamic programming technique to solve and analyse optimization problems in diverse application domains.*	**04**

	Contents Dynamic programming, advantages and limitations of dynamic programming. **Task 3:** Explore and compare alternative algorithmic design techniques to solve a complex problem such as the travelling salesman problem. Identify the advantages and limitations of the dynamic programming approach and implement dynamic programming approach for the given problem and analyze its performance. ***Learning Outcome:*** *A learner will be able to* *LO 3.1: Systematically evaluate a given problem to identify appropriate algorithm to solve it. Explore and compare alternative algorithmic design methods such as greedy, dynamic programming to select the best method for design and implementation of complex problems such as the travelling salesman problem. Use appropriate tools and techniques to solve the problem, and to analyze its time complexity. Identify the limitations of the dynamic programming approach.* *(P.I.-1.3.1)(P.I.-1.4.1) (P.I.-2.1.2)(P.I.-2.2.4)(P.I.-3.2.1) (P.I.-3.3.1)(P.I.-4.1.2) (P.I.-4.3.1)(P.I.-5.1.2)(P.I.-5.3.1)*	
04.	**Backtracking and Branch and bound** ***Learning Objective/s:*** *Learner is expected to apply and analyze Backtracking and Branch and bound algorithmic design techniques to solve a wide range of computational problems including optimization, constraint satisfaction, puzzle solving, graph algorithms etc.* **Contents** Backtracking and Branch and bound technique. Applications and limitations of Backtracking and Branch and bound. **Task 4**: Explore and compare alternative algorithmic design techniques to select best technique to solve a decision/optimization problem given in 4a, 4b. Implement the given problem using selected technique for example Backtracking, Branch and bound and analyze its time complexity. Identify the limitations of the selected approach. 4a. N-Queens problem or graph colouring problem. 4b. Traveling Salesman Problem or the 0/1 Knapsack problem. ***Learning Outcome:*** *A learner will be able to* *LO 4 .1: Systematically analyse and evaluate the requirements of a given problem to identify an appropriate algorithm to solve it. Explore and compare alternative algorithmic design methods, and select an optimal design scheme to solve a given problem. Apply different algorithmic design techniques such as Backtracking, Branch and Bound to design and implement effective solutions for the given problem.* *(P.I.-1.3.1) (P.I.-1.4.1) (P.I.-2.1.2) (P.I.-2.2.4) (P.I.-3.2.1) (P.I.-3.3.1) (P.I.-4.1.2) (P.I.-4.3.1)(P.I.-5.1.2)(P.I.-5.3.1)(P.I.-9.1.2)(P.I.-9.3.1)*	**08**
05.	**String matching and Advanced Algorithms** ***Learning Objective/s:*** *Learner is expected to apply and analyze string matching algorithms for identifying patterns in textual data. Learner is also expected to apply advance algorithms for solving compute intensive and optimization problems.*	**06**

	Contents String matching algorithms and Advanced optimization algorithms. **Task 5:** Evaluate the requirements of a given string matching problem to choose an appropriate algorithm to solve it. ***Learning Outcome:*** *A learner will be able to* *LO 5.1: Systematically evaluate the requirements of a given problem to identify and analyze appropriate algorithms and existing methods, explore algorithmic design alternatives, and apply suitable string matching algorithm/technique to solve a given string matching problem. (P.I.-1.4.1) (P.I.-2.2.3) (P.I.-3.2.1)(P.I.-4.1.2)(P.I.-5.1.2)* **Task 6:** Identify new algorithmic trends or tools relevant to genetic algorithms and apply them to select and implement an optimal design scheme for further development ***Learning Outcome:*** *A learner will be able to* *LO 5.2: Identify and apply the processes/algorithm to solve a given optimization problem. Identify new algorithmic trends or tools relevant to computational problem solving and applying these newly learned concepts in real-world or project scenarios to select an optimal design scheme for further development.* *(P.I.-1.3.1) (P.I.-2.1.2) (P.I.-3.3.1)(P.I.-4.3.1) (P.I.-5.3.1)(P.I.-11.2.1)(P.I.-11.3.3).*	
	Total	**30**

Performance Indicators :

P.I. No.	P.I. Statement
1.3.1	Apply engineering fundamentals.
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem.
2.1.2	Identifies processes/modules/algorithms of a computer-based system and parameters to solve a problem.
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions.
2.2.4	Compare and contrast alternative solution/methods to select the best methods.
2.4.2	Analyze and interpret the results using contemporary tools.
2.4.3	Identify the limitations of the solution and sources/causes.
3.2.1	Ability to explore design alternatives.
3.3.1	Ability to perform systematic evaluation of the degree to which several design concepts meet the criteria.
4.1.2	Ability to choose appropriate procedure/algorithm, data set and test cases.
4.3.1	Use appropriate procedures, tools and techniques to collect and analyse data.
5.1.2	Create/adapt/modify/extend tools and techniques to solve engineering problems.
5.3.1	Discuss limitations and validate tools, techniques and resources.
9.1.2	Produce clear, well-constructed, and well-supported written engineering documents.

9.3.1 Create engineering-standard figures, reports and drawings to complement writing and presentations.

11.2.1 Identify historic points of technological advance inengineering that required practitioners to seek education in order to stay current.

11.3.3 Apply the knowledge of technological advancements in engineering in order to stay current.

Course Outcomes:

Learner will be able to

1. Design an algorithm/procedure to solve the given problem using appropriate algorithmic design technique. (LO1.1, LO2.1 , LO 3.1, LO 4.1, LO 5.1, LO 5.2)
2. Implement an algorithm to solve the given problem using appropriate tool/technique (LO1.1, LO2.1 , LO 3.1, LO 4.1, LO 5.1, LO 5.2)
3. Analyse the complexity of different algorithms and draw conclusions. (LO1.1, LO2.1, LO 3.1, LO 4.1, LO 5.1, LO 5.2)

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSLBC403.1	3	3	3	3	3	-	-	-	3	-	3
CSLBC403.2	3	3	3	3	3	-	-	-	-	-	3
CSLBC403.3	3	3	3	3	3	-	-	-	3	-	3
Average	**3**	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**3**	**-**	**3**

Text Books:

1. Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein, 3rd Edition, 2009, MIT Press.
2. Fundamentals of Computer Algorithms, Horowitz, Sahani, Rajsekaran, 2nd Edition, 2009, Universities Press.

Reference Books:

1. Algorithms, Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, 1st Edition, 2006, McGraw- Hill India.
2. Design Methods and Analysis of Algorithm, S. K. Basu, 2nd Edition, 2005, PHI

Other Resources:

1. NPTEL Course: Design and Analysis of Algorithms By Prof. Madhavan Mukund, Chennai Mathematical Institute. Web link- https://nptel.ac.in/courses/106/106/106106131/
2. NPTEL Course: Data Structures and Algorithms by Prof. Naveen Garg, Department of Computer Science and Engineering Department, IIT Delhi:
Web link-https://nptel.ac.in/courses/106/102/106102064/

A. CONTINUOUS ASSESSMENT (25 MARKS)

Suggested breakup of distribution

a) Task Execution :10 Marks
Students will be given 8 tasks based on list as mentioned in the syllabus. Each task carries 10 Marks. Average will be taken of all tasks. Students are expected to identify and apply appropriate algorithm design techniques to solve a given problem and analyse the time complexity of the implemented solution using appropriate technique. Students will be evaluated based on parameters such as selection of appropriate algorithm design technique, its implementation for solving the given problem and time complexity analysis.

b) Practical Test: 05 Marks

Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

c) Oral based on entire syllabus: 05 Marks

d) Regularity & Active participation: 05 Marks

B. END SEMESTER EXAMINATION (Practical and Oral Exam) (25 Marks)

Suggested breakup of distribution

a) Task Execution :10 Marks

Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

b) Documentation on algorithm design technique selected for the given task and time complexity: 05 Marks

c) Oral based on entire syllabus:10 Marks

Two examiners, one Internal and one External will do the evaluation.

Course Type	Course Code	Course Name	Credits
LBC	CSLBC404	LINUX LABORATORY	01

Examination Scheme		
Continuous Assessment	**End Semester Exam (ESE)**	**Total**
25	25	50

Pre-requisite:

1. ESL103: Programming Laboratory-I (C)

Program Outcomes addressed:

PO1: Engineering knowledge

PO2: Problem analysis

PO3: Design/Development of Solutions

PO4: Conduct Investigations of Complex Problems

PO5: Engineering Tool Usage

PO11: Life-Long Learning

Course Objectives:

1. To train learners with demonstration and installation of OS guide with basic commands.
2. To instruct learners about demonstration of file and directory management services using Linux commands.
3. To instruct learners about demonstration of user, memory and process management services using Linux commands.
4. To provide hands-on for learners for demonstration of basic shell script.
5. To provide hands-on for learners for demonstration of shell script for advance applications.
6. To prepare students to execute Perl and awk script.

M	Detailed Contents	Hrs
00.	**Course Introduction** This course discovers the significance of Linux operating System, an open-source operating system that offers security, customization, and cost-effectiveness. The Linux command line is a powerful tool that can be used to save time in a variety of ways. By using the command line, users can quickly navigate through the file system, search for files, create and delete directories, and even run programs. This course majorly delivers the command line interaction of user with OS.	**01**
01.	**Basic Utility Commands** ***Learning Objective:*** *Demonstration of Linux Installation and Basic Commands.* **Contents:**	**03**

	Utility Commands: echo, clear, exit, date, time, uptime, cal, cat, man, which, history, id, pwd, whoami, pr, lp, lpr, lpstat, lpq, lprm, cancel, mail. cal: The calendar, date: Displaying the system date, echo: Displaying message, printf: An alternative to echo, bc: The calculator, script: Recording your session, Email basics, mailx: The universal mailer, passwd: Changing your password, who: Who are the users? uname: Knowing your machine characteristics, tty: Knowing your terminal, stty: Displaying and setting terminal characteristics. **Task 1**: Execute a set of essential Linux utility commands to perform routine operations such as navigating the file system, managing files and directories, monitoring system performance, viewing process details, manipulating file permissions, and handling text data. ***Learning Outcomes:*** *A learner will be able to* *LO 1.1: Use utility commands to inspect and manage system resources, apply* and execute the *commands and navigate the Linux environment confidently using command-line tools to review system performance*, modify/update the tools required for execution. *(*P.I. 1.3.1, *P.I.- 1.4.1,* P.I. 2.1.2, *P.I. 2.2.3, P.I. 3.1.3,* P.I. 3.2.1, P.I. 5.1.1, P.I. 5.1.2*)* ***Self-Learning Topics:*** *Demonstrate the commands line OS installation, memory partitions.*	
02.	**File and Directory Management Commands** ***Learning Objective:*** *To understand and perform file and directory handling in Linux using command line interface.*	**06**
	Contents: The file: Ordinary file, Directory file, Device file, Filename, the parent-child relationship, UNIX file system tree, The Unix file system, the home directory, pwd: Checking your current directory, cd: Changing the current directory, mkdir: Making directories, rmdir: Removing directories, ls: Listing directory contents, Absolute pathnames, Relative pathnames Handling ordinary files, cat: Displaying and creating files, cp: Copying file, rm: Deleting files, mv: Renaming files, more: Paging output The lp subsystem: printing a file, file: knowing the filetypes wc: Counting lines, words and characters, od: Displaying data in octal, cmp: Comparing two files, comm: What is common, diff: Converting one file to other, gzip and gunzip: Compressing and decompressing files and folders. tar: The archival program, zip and unzip: Compressing and archiving together. Basic file attributes, ls -l: Listing file attributes, the -d option: Listing directory attributes. File ownership, File permissions, chmod: Changing file permissions, directory permission, Changing file ownership, chown: Changing file owner, chgrp: Changing group owner **Task 2:** Study and execute different Linux file permission commands to understand how access rights are assigned, modified, and managed for files and directories. ***Learning Outcomes:*** *A learner will be able to* *LO 2.1: Use file management and file permission commands to inspect and manage file and directory functions, apply commands and navigate the Linux environment confidently using command-line tools to review system performance by utilizing/analizing advance tools. (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.2, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I. 5.1.1, P.I. 5.1.2)* **Task 3:** Execute commands to create files, view their permissions, and use commands such as chmod, chown, and chgrp to change permission settings and ownership.	

	Learning Outcomes: *A learner will be able to* *LO 2.2: Use file management and directory commands to create, add contents, remove, empty, function, apply commands and navigate the Linux environment confidently using command-line tools to review system performance by utilizing/analizing advance tools. (*P.I. 1.3.1, *P.I.- 1.4.1,* P.I. 2.1.2, *P.I. 2.2.3, P.I. 3.1.3,* P.I. 3.2.1, P.I. 5.1.1, P.I. 5.1.2*)*	
	Self-Learning Topics: *Commands to automate the disk space management and user alert by OS.*	
03.	**Memory, Process and User Management Commands** ***Learning Objective:*** *To understand the tasks in memory and user management and user management and able execute relevant commands on CLI.*	**06**
	Contents: a) Execution of Memory Management Commands like free, /proc/meminfo, top, htop, df, du, vmstat, demidecode, sar, pagesize, etc. b) Execution of Process Management Commands like ps, pstree, nice, kill, pkill, killall, xkill, fg, bg, pgrep, renice, etc. c)Execution of User Management Commands like who, whoami, su, sudo, login, logout, exit, passwd, useradd/adduser, usermod, userdel, groupadd, groupmod, groupdel, gpasswd, chown, chage, chgrp, chfn, etc. **Task 4:** Study and execute different Linux memory management commands to check memory allocation, secondary memory management, free space management. VM management, memory statistics. ***Learning Outcomes:*** *A learner will be able to* *LO 3.1: Able to use and apply advance Linux commands to handle process management, interprocess-communication, deadlock management by utilizing and analyzing system performance with respect to modern operating system compatibility to high-end applications. (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.2, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I. 5.1.1, P.I. 5.1.2)* **Task 5:** Study and execute different Linux process management commands to check status of foreground and background process management, process priority updating, process killing and deadlock management. ***Learning Outcomes:*** *A learner will be able to* *LO 3.2: Able to use and apply advance Linux commands to handle memory statistics, allocation, utilization, permissions by utilizing and analyzing system performance with respect to modern operating system compatibility to high-end applications. (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.2, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I. 5.1.1, P.I. 5.1.2)* **Task 6:** Execute user management commands to create system users and users permissions, user mod change, create user group, set group/user permission, user ownership setting, user credential/ system credential management. ***Learning Outcomes:*** *A learner will be able to* *LO 3.3: Able to use and apply advance Linux commands to handle user permissions, user creation, deletion, user log, credential management by utilizing and analyzing system performance with respect*	

	to modern operating system compatibility to high-end applications. . (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.2, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I. 5.1.1, P.I. 5.1.2)	
	Self-Learning Topics: *User and group management using shell script.*	
04.	**Basic Shell Script** ***Learning Objective:*** *To understand Linux shell processing and syntax associated with it to design batch files.*	**04**
	Contents: Write a basic shell script: with syntax of read, write statements, loop and control statements to execute system level functions. **Task 7: Developing and executing fundamental shell scripts to automate routine computational tasks in the Linux operating system.** i. Write a shell script to perform arithmetic operations. ii. Write a shell script to calculate simple interest. iii. Write a shell script to determine largest among three integer numbers. iv. Write a shell script to determine a given year is leap year or not. v. Write a shell script to print multiplication table of given number using while statement. ***Learning Outcomes:*** *A Learner will be able to* i. *L.O.4.1: Apply batch execution mode to design/develop shell script for utilizing various kernel operations for managing computational resources for modern operating system compatibility with advance tools. (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.1, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I.4.1.2, P.I. 4.3.1, P.I. 5.1.1, P.I. 5.1.2, P.I. 11.3.1, P.I.11.3.2)*	
	Self-Learning Topics: *Execute shell script for array, structures and composite data types.*	
05.	**Shell script for File and User Management** ***Learning Objectives:*** *To understand Linux shell processing and syntax associated with it to design batch files which are used for user and file management operations.*	**06**
	Contents: Advance shell script: with file, directory and user management functions. **Task 8: Developing and executing advance shell scripts to perform user and file management in Linux operating system.** i. Write shell script to check logged-in user details and number of logged-in user. ii. Write a shell script to read and check if the directory / file exists or not, if not make the directory / file. iii. Write a shell script using grep command to find the number of words, characters, and words and lines in a file. iv. Write shell script to calculate total number of files and folder in specified folder. v. Write a shell script that creates a directory and a file within it. Insert the contents into the file and duplicate the source folder to the specified target location. ***Learning Outcomes:*** *A learner will be able to* *LO.5.1: Apply batch execution mode to design/develop advance shell script file for utilizing various kernel operations for managing user operation/ file management for better computational practices*	

	in modern operating system (e.g. Windows 11) and utilizing OS compatibility with advance tools. (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.1, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I.4.1.2, P.I. 4.3.1, P.I. 5.1.1, P.I. 5.1.2, P.I. 11.3.1, P.I.11.3.2)	
	Self-Learning Topics: *Code shell script for network and infrastructure security.*	
06.	**Execute Perl and Awk Script** ***Learning Objective:*** *To implement complex problems using data structures and algorithms, and analysing data structures inside data management models.*	**04**
	Contents Introduction and use of Perl and Awk script, Syntax in Perl, Syntax in awk, Control and conditional statements in Perl and Awk script. Perform Perl and AWK script for **Task 9:** Execute Perl script for utilizing file management operations Create write/ append/ count number of words/ copy file content to another file/ delete file using Perl script. ***Learning Outcomes*** *A learner will be able to* *LO.6.1: - Apply command line script execution to build skills in handling files, manipulating text, and processing large datasets efficiently using Perl for improving logical thinking and problem-solving abilities through implementation of modular and reusable script to gain proficiency in using scripting techniques for system administration, data processing, and tool development. (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.1, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I.4.1.2, P.I. 4.3.1, P.I. 5.1.1, P.I. 5.1.2, P.I. 11.3.1, P.I.11.3.2)* **Task 10:** Write AWK script for file management. Write AWK Script to Count Words in Each Line, AWK Script to Replace a Word, Print Number of Lines in a File, Print Each Line of a File, Calculate Sum of Numbers in a Column, Change Field Separator (e.g., CSV File) ***Learning Outcomes*** *A learner will be able to* *LO.6.2: - Apply command line script execution to build skills in handling files, manipulating text, and processing large datasets efficiently using AWK, for file processing, scan for patterns, and perform minor actions like formatting text etc. (P.I. 1.3.1, P.I.- 1.4.1, P.I. 2.1.1, P.I. 2.2.3, P.I. 3.1.3, P.I. 3.2.1, P.I.4.1.2, P.I. 4.3.1, P.I. 5.1.1, P.I. 5.1.2, P.I. 11.3.1, P.I.11.3.2)*	
	Self-Learning Topics: *Perl script to develop applications in open-source systems.*	
	Total	**30**

Performance Indicators:

P.I. No.	P.I. Statement
1.3.1	Apply engineering fundamentals
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem.
2.2.3	Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions.
3.1.3	Ability to review state of the art literature to synthesize system requirements.
3.2.1	Ability to explore design alternatives.
4.1.2	Ability to choose appropriate procedure/algorithm, data set and test cases.
4.3.1	Use appropriate procedures, tools and techniques to collect and analyze data.

4.3.3 Represent data (in tabular and/or graphical forms) so as to facilitate analysis and explanation of the data, and drawing of conclusions.

5.1.1 Identify modern engineering tools, techniques and resources for engineering activities

5.1.2 Create/adapt/modify/extend tools and techniques to solve engineering problems

5.3.1 Discuss limitations and validate tools, techniques and resources.

11.3.1 Source and comprehend technical literature and other credible sources of information.

11.3.2 Analyze sourced technical and popular information for feasibility, viability, sustainability, etc.

Course Outcomes:

Learner will be able to

1. *Use the basic utilities commands for OS installation. (LO 1.1)*
2. *Apply basic Linux commands for file management, process control, and system navigation. (LO 2.1, LO 2.2)*
3. *Configure and manage user accounts, system memory, processes in a Linux environment. (LO 3., LO 3.2, LO 3.3)*
4. *Develop and execute shell scripts to automate administrative and computational tasks. (LO 4.1, LO 5.1)*
5. *Develop and execute Perl and Awk scripting to demonstrate your proficiency in text processing, pattern matching, file handling, and data manipulation. (LO 6.1, LO 6.2)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSLBC404.1	3	3	3	-	3	-	-	-	-	-	-
CSLBC404.2	3	3	3	-	3	-	-	-	-	-	-
CSLBC404.3	3	3	3	-	3	-	-	-	-	-	-
CSLBC404.4	3	3	3	3	3	-	-	-	-	-	3
CSLBC404.5	3	3	3	3	3	-	-	-	-	-	3
Average	**3**	**3**	**3**	**3**	**3**	-	-	-	-	-	**3**

Text Books:

1. Unix Concepts and Applications, S. Das, 4th edition., 2017, McGraw Hill.
2. Mastering Unix Shell Scripting, R. Michael, 2nd edition, 2008, Wiley.
3. Linux Experiments and Open-Source Technologies, D. Ambawade, D. Shah, 2nd edition, 2014, Dreamtech Press.

Reference Books:

1. Unix Shell Programming, Y. Kanetkar, 3rd edition, 2003, BPB Publications.
2. Unix and Shell Programming, B. Forouzan and R. Gilberg, 4th edition, Cengage Learning, 2003.

Other Resources:

1.

2. Digital material:
Web Link- Install Ubuntu desktop | Ubuntu
Web Link- The Linux Command Handbook – Learn Linux Commands for Beginners (freecodecamp.org)

A. CONTINUOUS ASSESSMENT (25 MARKS)

Suggested breakup of distribution

a) Command based script execution :10 Marks

Students will be given 10 tasks based on list as per mentioned in the syllabus. Each task carries 10 Marks. Average will be taken of all tasks. Students are expected to execute the basic utility and advanced commands of Linux operating system also they have to perform command-based shell, Perl, AWK script. Students will be evaluated based on logic building for the given script and expected output.

b) Practical Test: 10 Marks

Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

c) Regularity & Active participation: 05 Marks

B. END SEMESTER EXAMINATION (Practical and Oral Exam) (25 Marks)

Suggested breakup of distribution

a) Command based script **e**xecution: 10 Marks

Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

b) Results and discussion, Inferences drawn: 05 Marks

c) Oral based on entire syllabus :10 Marks

Two examiners, one Internal and one External will do the evaluation.

Course Type	Course Code	Course Name	Credits
LBC	CSLBC405	SOFTWARE DEVELOPMENT LABORATORY	01

Examination Scheme		
Continuous Assessment	End Semester Exam(ESE)	Total
25	25	50

Pre-requisite:

1. ESL103: Programming Laboratory-I (C)
2. ESL205: Programming Laboratory-II (Java)
3. CSSBL301 : Python Laboratory

Program Outcomes addressed:

1. PO 1: Engineering knowledge
2. PO 2: Problem analysis
3. PO 3: Design/Development of Solutions
4. PO 4: Conduct investigations of complex problems
5. PO 5: Engineering tool usage
6. PO 7: Ethics
7. PO 8: Individual and collaborative team work
8. PO 9: Communication
9. PO 10: Project management and finance
10. PO 11: Life-long learning

Course Objectives:

1. To familiarize leaners, the basics of software engineering principles.
2. To make learner understand documentation Requirements, analysis, planning and scheduling.
3. To acquaint leaners with skills to design test case plans for testing software using dynamic testing techniques.
4. To make learner able to apply knowledge of automation testing.
5. To make leaner infer and apply quality assurance and management.

Module	Detailed Contents	Hrs
00.	**Course Introduction** This course deals with basics of software engineering principles, phases of software development life cycle, various process models which will be helpful to follow the process during software development, identify early challenges or cost constraints to know team's development process for the future. Software Testing tools are used in industries for automating the entire testing process. Software QA and testing helps prevent defects from reaching end-users. By detecting issues earlier in software development cycles through extensive testing, problems can be fixed before software gets deployed.	-

01.	**Introduction to concepts of software engineering principles and software development process.** ***Learning Objective/s:*** *To infer basics of software engineering principles and software development life cycle and apply it for the project development process.* *Learner is expected to gain knowledge of process models and also expected to determine the right process followed for software development.*	**06**
	Contents Software Development Life Cycle. Software Development Life Cycle models. **Task 1:** Select a problem statement and write case study on Software Development Life Cycle for the same. Compare different SDLC models. Implement Waterfall Model and Agile model for the selected problem statement. Analyze the difference between waterfall model and agile model. Select and document the most appropriate model for effective project development. ***Learning Outcome :*** *A learner will be able to* *LO 1.1: Formulate a software engineering problem, identify the tasks required to complete the software development and apply software engineering principles to document the phases of the software development life cycle. Illustrate and compare design alternatives and process models based on economic and financial considerations. Select and document the most appropriate and optimal process model for effective project development. (P.I.-1.3.1) (P.I.-1.4.1) (P.I.-2.1.1) (P.I.-2.2.5) (P.I.-3.2.1) (P.I.-3.4.1) (P.I.- 9.1.2) (P.I.- 9.3.2) (P.I.-10.2.1) (P.I.-10.3.1).* *1.*	
02.	**Software Requirement Analysis and scheduling** ***Learning Objective/s:*** *To paraphrase and apply knowledge of documentation requirements, analysis, planning and scheduling for software development.*	**06**
	Contents Requirements analysis. Software Requirement Specification (SRS). Work breakdown structure (WBS) **Task 2:** Identify the system requirements for the selected problem statement and write a Software Requirement Specification (SRS). ***Learning Outcome:*** *A learner will be able to* *LO 2.1: Apply the basics of software engineering to identify and analyze the objectives, system requirements and parameters to solve a real-world problem. Prepare and document a comprehensive Software Requirements Specification (SRS) for the selected problem statement. (P.I.-1.3.1, P.I.-2.1.2, P.I.-3.1.2, P.I.-3.1.6)* **Task 3:** Create a Work breakdown structure (WBS) for the selected problem statement and use appropriate tools to prepare Gantt chart for the selected problem statement. ***Learning Outcome:*** *A learner will be able to* *LO 2.2: Analyse a real-world problem to identify functionalities for creating a Work Breakdown Structure (WBS), use WBS to schedule and allocate various tasks during the SDLC, develop Gantt charts and other timeline charts using appropriate tools and*	

	techniques, and identify the limitations of these tools in maintaining the software development timeline. (P.I.-1.4.1, P.I.-2.2.2, P.I.-5.1.2, P.I.-5.3.1)	
03.	**Software Testing Techniques** ***Learning Objective/s:*** *To illustrate and apply black box testing techniques for designing test cases to identify and analyze structural errors. Learner is expected to infer knowledge of white box testing techniques and use white box testing techniques to ensure that all the internal components work as expected and are error-free.*	**08**
	Contents Black box testing. White box testing. **Task 4:** Design test cases using black box testing techniques for the selected project, execute the test cases and discuss the results. ***Learning Outcome :*** *A learner will be able to* *LO 3.1: Apply black box testing techniques such as Equivalence Partitioning, Boundary Value Checking (BVC) etc. to identify processes and algorithms for error detection. Explore, compare and select, apply best testing technique for the selected problem statement. Apply state-table–based testing to identify the state of bugs, determine system functionalities and validate designed test cases. (P.I.-1.3.1, P.I.-1.4.1, P.I.-2.1.2, P.I.-2.2.4, P.I.-3.2.1, P.I.-3.4.3)* **Task 5:** Design test cases using path testing- a white box testing technique and execute test cases for the same selected problem statement to identify and fix the logical errors. ***Learning Outcome :*** *A learner will be able to* *LO 3.2: Choose appropriate procedures, algorithms, datasets, and test cases to identify structural errors, and analyze structural and logical bugs by running and refining test cases over time. (P.I.-4.1.2, P.I.-4.3.2)*	
04.	**Automation and Testing Tools** ***Learning Objective/s:*** *To infer knowledge of easy approach of testing using automated tools and also expected to select and use automation tools which mainly focuses on reducing manual human activity. Learner is expected to summarize effectiveness, and coverage of software testing to save time and effort using automated tools. Also expected to compare the actual output against the expected outcome using testing tool.*	**06**
	Contents Automation tools for and Testing, Bug tracking. **Task 6**: Study and categorize various testing tools and implement anyone open-source testing tool. ***Learning Outcomes:*** *A learner will be able to* *LO 4.1: Identify, categorize, and apply automation tools for static and dynamic testing; use appropriate testing tools and techniques to design test data and test cases for various modules; analyze and interpret test cases and test results using contemporary tools; evaluate the credibility, accuracy, and limitations of these tools and automation processes; and explore new tools and techniques to collect, analyze, and refine test data*	

	over time. (P.I.-1.4.1, P.I.-3.2.2, P.I.-3.4.2, P.I.-4.3.1, P.I.-4.3.2, P.I.-5.1.2, P.I.-11.2.2, P.I.-2.4.2, P.I.-2.4.3, P.I.-5.3.2, P.I.-11.1.2) **Task 7:** Study and implementation of anyone bug tracking tool. ***Learning Outcomes:*** *A learner will be able to* *LO 4.2: Identify and apply automation tools for bug tracking, and use appropriate testing tools and techniques to collect and analyze test data. Analyze and interpret test results using contemporary tools; (P.I.-1.3.1, P.I.-2.4.2, P.I.-4.3.1)*	
05	**Software quality Assurance** ***Learning Objective/s:*** *To summarize basics of software quality assurance and management. Also expected to use and analyse various quality factors and criteria which helps to prevent defects from reaching end-users*	**04**
	Contents Software quality management and assurance. McCall's quality factors and criteria. Software development standards such as ISO 9000:2000 and Six Sigma. **Task 8:** Identify Quality Attributes and Relationships for the various properties used for the selected problem statement. ***Learning Outcomes:*** *A learner will be able to* *LO 5.1: Identify software quality attributes and apply software quality management and assurance to determine possible solutions; apply engineering principles to formulate system modules using quality attributes based on McCall's quality factors and criteria; and analyze quality attributes and criteria for software development using standards such as ISO 9000:2000 and Six Sigma. (P.I.-1.3.1, P.I.-7.1.1, P.I.-1.4.1, P.I.-2.3.1, P.I.-7.2.2, P.I.-8.2.1, P.I.-9.2.1, P.I.-2.3.2, P.I.-8.3.1, P.I.-9.1.2)*	
	Total	**30**

Performance Indicators :

P.I. No.	P.I. Statement
1.3.1	Apply engineering fundamentals
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem
2.1.1	Evaluate problem statements and identifies objectives
2.1.2	Identifies processes/modules/algorithms of a computer-based system and parameters to solve a problem.
2.2.2	Identifies functionalities and computing resources.
2.2.4	Compare and contrast alternative solution/methods to select the best methods
2.2.5	Compare and contrast alternative solution processes to select the best process.
2.3.1	Able to apply computer engineering principles to formulate modules of a system with required applicability and performance
2.3.2	Identify design constraints for required performance criteria

2.4.2	Analyze and interpret the results using contemporary tools
2.4.3	Identify the limitations of the solution and sources/causes.
3.1.2	Able to identify and document system requirements from stake holders
3.1.6	Ability to develop software requirement specifications (SRS).
3.2.1	Ability to explore design alternatives.
3.2.2	Ability to produce a variety of potential design solutions suited to meet functional requirements.
3.4.1	Ability to refine architecture design into a detailed design within the existing constraints.
3.4.2	Ability to implement and integrate the modules.
3.4.3	Ability to verify the functionalities and validate the design.
4.1.2	Ability to choose appropriate procedure/algorithm, data set and test cases.
4.3.1	Use appropriate procedures tools and techniques to collect and analyze data
4.3.2	Critically analyse data for trends and correlations, stating possible errors and limitations
5.1.2	Create/adapt/modify/extend tools and techniques to solve engineering problems.
5.3.2	Verify the credibility of results from tool use with reference to the accuracy and limitations, and the assumptions inherent in their use.
7.1.1	Identify situations of unethical professional conduct and propose ethical alternatives.
7.2.2	Examine and apply moral & ethical principles to known case studies.
8.2.1	Demonstrate effective communication, problem solving, conflict resolution and leadership skills
8.3.1	Present results as a team, with smooth integration of contributions from all individual efforts.
9.1.2	Produce clear, well-constructed, and well-supported written engineering documents
9.2.1	Listen to and comprehend information, instructions, and viewpoints of others
9.3.2	Use a variety of media effectively to convey a message in a document or a presentation
10.2.1	Analyze and select the most appropriate proposal based on economic and financial considerations.
10.3.1	Identify the tasks required to complete an engineering activity, and the resources required to complete the tasks.
11.1.2	Identify deficiencies or gaps in knowledge and demonstrate an ability to source information to close this gap
11.2.2	Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field

Course Outcomes:

Learner will be able to

1. Use concepts of software engineering process models to optimize the flow of project development process. *(LO 1.1)*
2. Identify requirements, analyse, plan, schedule and track the progress of the projects. *(LO 2.1, LO 2.2)*

3. Use various software testing methods and strategies to identify bugs. *(LO 3.1, LO 3.2)*
4. Use of automation tools. *(LO 4.1, LO 4.2)*
5. Apply the software quality assurance factors to Identify Quality Attributes and their Relationships. *(LO 5.1)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSLBC405.1	3	3	3	-	-	-	-	-	3	3	-
CSLBC405.2	3	3	3	-	3	-	-	-	-	-	-
CSLBC405.3	3	3	3	3	-	-	-	-	-	-	-
CSLBC405.4	3	3	3	3	3	-	-	-	-	-	3
CSLBC405.5	3	3	-	-	-	-	3	3	3	-	-
Average	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

Text Books:

1. Software Engineering: A practitioner's approach, Roger S. Pressman, 7 th edition ,2010, McGraw Hill
2. Fundamentals of Software Engineering, Rajib Mall, 7th edition ,2014, Prentice Hall India
3. Software Testing Principles and Practices, Naresh Chauhan, 2nd edition, 2016, Oxford Higher Education
4. Software Testing and quality theory and practice, Kshirasagar Naik, Priyadarshi Tripathy, 1st edition 2008,Wiley Publication.

Reference Books:

1. An integrated approach to Software Engineering, Pankaj Jalote, , 3rd edition 2005, Springer/Narosa.
2. Effective Methods for Software Testing, Willam E. Perry, third edition, 2006,Wiley Publication.
3. Software Testing Concepts and Tools, Nageswara Rao Pusuluri, ISE Edition, 2006, Dreamtech press

Other Resources:

1. NPTEL Course: Introduction to software Engineering By Prof. N.L.Sarda, Department of Computer Science and Engineering, IIT Bombay
   Web link- https://archive.nptel.ac.in/courses/106/101/106101061/
2. Course on Software Testing By Prof. Rajib mall, Department of Computer Science and Engineering, IIT Kharagpur
   Web link- https://nptel.ac.in/courses/106105150/.

A. CONTINUOUS ASSESSMENT (25 MARKS)

Suggested breakup of distribution

a) Task Execution :10 Marks
Students will be given 8 tasks based on list as per mentioned in the syllabus. Each task carries 10 Marks. Average will be taken of all tasks. Students are expected to identify and apply

software engineering principles also perform manual and automated testing to find the logical and structural errors and also apply quality assurance factors and criteria. Students will be evaluated based on parameters such as identification of process models, testing techniques and quality check.

b) Practical Test: 05 Marks

Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

c) Oral based on entire syllabus: 05 Marks

d) Regularity & Active participation: 05 Marks

B. END SEMESTER EXAMINATION (Practical and Oral Exam) (25 Marks)

Suggested breakup of distribution

a) Task Execution :10 Marks
Students will be given task (different for each student) and they will be evaluated based on the parameters mentioned in continuous assessment.

b) Documentation on Software Development Life Cycle: 05 Marks

c) Oral based on entire syllabus:10 Marks

Two examiners, one Internal and one External will do the evaluation.

Course Type	Course Code	Course Name	Credits
SBL	CSSBL402	FULL STACK DEVELOPMENT LABORATORY	02

Examination Scheme		
Continuous Assessment	**End Semester Exam (ESE)**	**Total**
50	50	100

Pre-requisite:

1. ESL205: Programming Laboratory-II (Java)
2. CSSBL301: Python Laboratory

Program Outcomes addressed:

1. PO1: Engineering knowledge
2. PO2: Problem analysis
3. PO3: Design/Development of solution
4. PO5: Engineering tool usage
5. PO8: Individual and Collaborative team work
6. PO9: Communication
7. PO11: Life-long learning

Course Objectives:

1. To prepare students with the skills to design and develop web applications using HTML and multimedia elements.
2. To make students with the skills to develop and style web applications using HTML and CSS, focusing on creating responsive and interactive user interfaces.
3. To equip students with the skills to analyze and interpret programming results using JavaScript techniques.
4. To train students with the skills to evaluate and implement modern front-end technologies, focusing on React and AngularJS.
5. To equip students with the skills to install, configure, and develop applications using Node.js, Express, and MongoDB.

Module	Detailed Contents	Hrs
00.	**Course Introduction** In the Full Stack Web Development Lab, students explore modern web development using React and Angular. They gain practical experience building dynamic web applications from frontend to backend.	**01**
01.	**HTML5** ***Learning Objective/s:*** *To demonstrate and apply HTML tags to develop webpages with well-structured HTML. Also expected to create web pages using text formatting, graphics, audio, and video elements.*	**03**

	Contents: Elements, Attributes, Head, Body, Hyperlink, Formatting, Images, Tables, List, Frames, Forms, Multimedia. Scenario: Building a "University Club Portal" Homepage You are a web developer tasked with creating the homepage for your university's "Digital Innovators Club." This page will serve as the central hub for information and member engagement. Task 1: Build a semantically structured, multi-section HTML5 page that incorporates all the specified elements to create a functional and informative club portal.	
	Self-Learning Topics: *Canvas and SVG Graphic, Parallax Scrolling*	
	Learning Outcomes : *A learner will be able to* *LO 1.1: Apply HTML elements and attributes to structure web pages, utilize head and body tags to organize content, and integrate hyperlinks to connect web resources, ensuring a comprehensive understanding of web page formatting and structure. (P.I.- 1.3.1, 3.4.2, 5.1.1, 8.1.1)* *LO 1.2: Implement multimedia elements such as images, frames, and forms to enhance user interaction, demonstrate proficiency in creating and managing tables and lists for data presentation, and collaborate effectively in teams to develop and debug web applications, ensuring seamless integration of multimedia components. (P.I.- 1.4.1, 3.4.3, 5.2.2, 8.3.1)*	
02.	**CSS3** ***Learning Objective/s:*** *To apply CSS in styling tables and lists, also use advanced selectors to precisely target and style specific elements within a document.*	**04**
	Contents: Syntax, Inclusion, Color, Background, Fonts, Tables, lists, CSS3 selectors, Pseudo classes, Pseudo elements Scenario: Styling and Enhancing the "University Club Portal" The Digital Innovators Club leadership has reviewed the HTML prototype and now wants a visually appealing, branded, and user-friendly interface. Your task is to apply CSS to transform the raw HTML into a polished website. Task 2: Using the HTML file you created in Task 1, create an external CSS stylesheet to style all elements of the Club Portal homepage according to a defined brand guideline, implementing advanced CSS3 features for a modern look and feel.	
	Self-Learning Topics: *CSS Best Practices and Optimization*	
	Learning Outcomes :	

	A learner will be able to *LO 2.1: Apply HTML syntax and inclusion techniques to develop web pages, utilize color and background properties to enhance visual appeal, recognize the strengths and limitations of various font styles, and read, understand, and interpret technical documentation related to HTML and CSS. (P.I.- 1.3.1, 2.1.2, 5.2.1, 9.1.1)* *LO 2.2: Apply CSS3 selectors, pseudo-classes, and pseudo-elements to style web pages, identify and implement tables and lists for data organization, demonstrate proficiency in using CSS for complex styling tasks, and demonstrate effective communication by producing well-structured CSS design documentation. (P.I.- 1.4.1, 2.3.2, 5.2.2, 9.1.2)*	
03.	**Bootstrap** ***Learning Objective/s:*** *To create responsive, mobile-first websites and to ensure all interface elements of a website work optimally on all screen sizes.*	**04**
	Contents: Grid system, Forms, Button, Navbar, Breadcrumb, Jumbotron Scenario: Responsive Redevelopment with Bootstrap The Club Portal is gaining traction, and the leadership has noticed that many members access it from their phones and tablets. The current design does not scale well. Your new task is to refactor the entire portal using Bootstrap to create a fully responsive, mobile-first, and professionally consistent website. Task 3: Rebuild the "University Club Portal" homepage using the Bootstrap framework. You will replace your custom CSS and structural HTML with Bootstrap's grid system and pre-built components to ensure an optimal viewing experience across all device sizes.	
	Self-Learning Topics: *Customizing Bootstrap Themes*	
	Learning Outcomes: *A learner will be able to* *LO 3.1: Implement and integrate grid systems and forms within a web application, identify and apply HTML and CSS tools to create responsive and interactive user interfaces, and evaluate the limitations of these tools to validate their effectiveness in real-world applications. (P.I.- 3.4.2, 5.1.1, 5.3.1)* *LO 3.2: Design and develop interactive UI components by integrating buttons, navbars, breadcrumbs, and jumbotrons, and evaluate the limitations of these components, validating their effectiveness in real-world applications. (P.I.- 3.4.3, 5.3.1)*	
04.	**JavaScript** ***Learning Objective/s:*** *To recall and apply JavaScript and its concepts and develop understanding behind JavaScript applications.*	**16**

	Contents: Variables, Operators, Conditions, Loops, Functions, Events, Classes and Objects, Error handling, Validations, Arrays, String, Date Scenario: Adding Dynamic Blog Management with JavaScript The Club Portal needs a dynamic "Blog" section where members can post updates without editing the HTML. Your task is to implement a client-side blog manager using JavaScript, transforming the static site into an interactive application. Task 4: You will build a Blog Management System on a new page (blog-manager.html). This system will use JavaScript to handle all core programming concepts, allowing users to view, add, and delete blog posts entirely in the browser. ***Self-Learning Topics:*** *Testing and Test-Driven Development, Web APIs and Browser Features* ***Learning Outcomes:*** *A learner will be able to* *LO 4.1: Analyze and interpret programming results using contemporary tools, identify the strengths and limitations of variables, operators, and conditions, and demonstrate proficiency in using discipline-specific tools for handling loops, functions, and events in programming. (P.I.- 2.4.2, 5.2.1, 5.2.2)* *LO 4.2: Identify the limitations of various programming constructs such as classes, objects, and error handling, and demonstrate proficiency in using discipline-specific tools for validations, arrays, strings, and date manipulations. (P.I.- 2.4.3, 5.2.2)*	
05.	**React and Angular** ***Learning Objective/s:*** *To recognize the steps involved in setting up and configuring a React and Angular development environment to ensure a smooth setup. Design and develop an understanding behind React apps and Angular apps.* **Contents:** Installation and Configuration. JSX, Components, Props, State, Forms, Events, Routers, Refs, Keys. Overview of AngularJS, Need of AngularJS in real web sites, AngularJS modules, AngularJS built-in directives, AngularJS custom directives, AngularJS expressions, Angular JS Data Binding, AngularJS filters, AngularJS controllers, AngularJS scope, AngularJS dependency injection, Angular JS Services, Form Validation, Routing using ng-Route, ng-Repeat, ng-style, ng-view, Built-in Helper Functions, Using Angular JS with Typescript Scenario: Modernizing the Blog Manager with React The JavaScript-based blog manager works but is slow and difficult to maintain as features grow. The club has decided to modernize the frontend using React to create a faster, more structured, and maintainable Single-Page Application (SPA). Task 5: Rebuild the client-side Blog Management System as a React application. You will set up a new project, create components, and manage	**20**

	state to replicate and enhance the functionality from the previous JavaScript task. Scenario: Building a Student Records Manager with Angular The university is impressed with the club portal and wants a separate, robust internal application for managing student records. For this project, you will use Angular to build a "Student Records Manager," leveraging its powerful built-in features for data binding and form validation. Task 6: Create a new Angular application that allows users to view, add, and validate student records.	
	Self-Learning Topics: *Context API and Redux, State Management with NgRx (Optional), Testing and Debugging*	
	Learning Outcomes: *A learner will be able to* *LO 5.1: Evaluate and implement React components, props, and state management in web applications, recognize the importance of lifelong learning in adapting to evolving front-end technologies, and analyze sourced technical information for the feasibility and sustainability of React integration. (P.I.- 3.4.2, 5.1.1, 11.1.1, 11.3.2)* ***LO 5.2: Verify functionalities and validate the integration of AngularJS services, form validation, and routing, and recognize the importance of lifelong learning in adapting to evolving front-end technologies. (P.I.- 3.4.3, 5.2.2, 11.1.1)***	
06.	**Node.js** ***Learning Objective/s:*** *To apply callback functions and event loops, also able to install and configure Express with ease, and build a working Express application on their own to showcase server-side JavaScript coding proficiency. Also expected to build REST API using MongoDB.*	**12**
	Contents: Installation and Configuration, Callbacks, Event loops, creating express app, create Mongodb application Scenario: Full-Stack Integration with Node.js, Express, and MongoDB The Blog Manager (React) and Student Records Manager (Angular) currently lose all data on refresh because they rely on client-side storage. The final project is to build a secure, scalable backend API with a database. This will allow the frontend applications to persist data permanently and enable features like user authentication in the future. Task 7: You will build a RESTful API using Node.js, Express, and MongoDB. This API will serve as the backend for your React Blog Manager application from Task 6. Finally, you will integrate the React frontend with your new backend to create a complete, functioning full-stack application. Task 8: Mini Project - Full-Stack Integration	
	Self-Learning Topics: *Template Engines and Views, Indexes and Query Optimization*	
	Learning Outcomes: *A learner will be able to*	

	LO 6.1: Identify modern engineering tools for the installation and configuration of Node.js applications, recognize and adapt to diverse working and learning preferences, read, understand, and interpret documentation for callbacks and event loops, and demonstrate proficiency in using discipline-specific tools for creating Express and MongoDB applications. (P.I.- 5.1.1, 5.2.2, 8.1.1, 9.1.1) *LO 6.2: Present application development results as a team with well-integrated contributions from all members, produce clear, well-constructed, and well-supported documentation for Express and MongoDB applications, and demonstrate proficiency in using discipline-specific tools for creating and managing these applications. (P.I.- 5.2.2, 8.3.1, 9.1.2, 5.2.2)*	
	Total	**60**

Performance Indicators:

P.I. No.	**P.I. Statement**
1.3.1	Apply engineering fundamentals
1.4.1	Apply theory and principles of computer science engineering to solve an engineering problem
2.1.1	Evaluate problem statements and identifies objectives
2.1.2	Identifies processes/modules/algorithms of a computer based system and parameters to solve a problem
2.2.4	Compare and contrast alternative solution/methods to select the best methods
2.4.2	Analyze and interpret the results using contemporary tools.
2.4.3	Identify the limitations of the solution and sources/causes.
3.4.2	Ability to implement and integrate the modules.
3.4.3	Ability to verify the functionalities and validate the design.
5.1.1	Identify modern engineering tools, techniques and resources for engineering activities
5.2.1	Identify the strengths and limitations of tools for (i) acquiring information, (ii) modelling and simulating, (iii) monitoring system performance, and (iv) creating engineering designs.
5.2.2	Demonstrate proficiency in using discipline specific tools
5.3.1	Discuss limitations and validate tools, techniques and resources
8.1.1	Recognize a variety of working and learning preferences; appreciate the value of diversity on a team
8.3.1	Present results as a team, with smooth integration of contributions from all individual efforts
9.1.1	Read, understand and interpret technical and nontechnical information
9.1.2	Produce clear, well-constructed, and well-supported written engineering documents
11.1.1	Describe the rationale for requirement for continuing professional development
11.3.2	Analyze sourced technical and popular information for feasibility, viability, sustainability, etc.

Course Outcomes:

Learner will be able to

1. To structure web pages, integrate multimedia components, and collaborate effectively in web application development. *(LO 1.1, LO 1.2)*
2. Apply HTML and CSS techniques to build and style web pages, integrate various UI components, and effectively communicate their design through well-structured documentation. *(LO 1.1, LO 1.2, LO 2.1, LO 2.2, LO 3.1, LO 3.2)*
3. Identify the strengths and limitations of JavaScript programming constructs and demonstrate proficiency in using discipline-specific tools for handling loops, functions, events, validations, arrays, strings, and date manipulations. *(LO 4.1, LO 4.2)*
4. Evaluate and implement React components and state management, and verify and validate AngularJS services, form validation, and routing, while recognizing the importance of lifelong learning in adapting to evolving technologies. *(LO 5.1, LO 5.2)*
5. Identify and use modern engineering tools for Node.js applications, collaborate effectively in teams, and produce well-constructed documentation for Express and MongoDB applications. *(LO 5.1, LO 5.2, LO6.1, LO6.2)*

CO-PO Mapping Table with Correlation Level

CO ID	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CSSBL402.1	3	-	3	-	3	-	-	3	-	-	-
CSSBL402.2	3	3	3	-	3	-	-	3	3	-	-
CSSBL402.3	-	3	-	-	3	-	-	-	-	-	-
CSSBL402.4	-	-	3	-	3	-	-	-	-	-	3
CSSBL402.5	-	-	3	-	3	-	-	3	3	-	3
Average	**3**	**3**	**3**	**-**	**3**	**-**	**-**	**3**	**3**	**-**	**3**

Text Books:

1. HTML & CSS: Design and Build Web Sites, Jon Duckett,, First Edition, 2011, Wiley.
2. HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery) 2Ed., DT Editorial Services, Second Edition, 2016, Dreamtech Press.
3. Learning React Functional Web Development with React and Redux, Alex Banks and Eve Porcello, First Edition, 2017, Shroff/O'Reilly O'Reilly
4. Learning Node.js Development, Andrew Mead, Kindle Edition, 2018, Packet Publishing

Reference Books:

1. JavaScript & jQuery: Interactive Front-End Development, Jon Duckett, First edition, 2014, Wiley .
2. Full Stack Web Development For Beginners, Riaz Ahmed, 2021
3. Beginning Node.js, Express & MongoDB Development, Greg Lim, 2020
4. Full-Stack React Projects: Modern web development using React 16, Node, Express, and MongoDB, Shama Hoque, First edition, 2018, Packt Publication

Other Resources:

1. JavaScript Tutorial

   Web link: https://www.w3schools.com/js/
2. React: The library for web and native user interfaces

   Web link: https://react.dev/

3. Deliver web apps with confidence

   Web link: https://angular.io/
4. Run JavaScript Everywhere

   Web link: https://nodejs.org/en
5. Express4.18.3 Fast, unopinionated, minimalist web framework for Node.js

   Web link: https://expressjs.com/
6. MongoDB

   Web link: https://www.mongodb.com/

A. CONTINUOUS ASSESSMENT (50 MARKS)

1. Task Execution, Performance (20 Marks)

Students will be evaluated based on following:

i. Logic building for the given task (06 marks)
ii. Rectifying logical errors and syntax errors (04 marks)
iii. Well-structured and organized program (04 marks)
iv. Effective use and integration of the various technologies involved in full stack development, including React, Angular, MongoDB, Express, and Node.js. (06 marks)

2. Regularity and active Participation (05 Marks)

3. Mini Project Evaluation (25 Marks)

A group of 3-4 students should be assigned a real life problem statement. Evaluation will focus on the project's ability to meet functional requirements, such as CRUD operations, user authentication, and data validation, ensuring that the application behaves as intended.

Attention will be given to the usability and intuitiveness of the application's interface, including navigation, responsiveness, and feedback mechanisms, to ensure a positive user experience.

Assessment will consider the quality of the codebase, including adherence to coding standards, modularity, reusability, and readability, promoting maintainability and scalability of the project.

Projects will be evaluated for performance optimization techniques, such as minimizing load times, reducing server requests, and efficient database queries, to ensure responsiveness and efficiency.

Evaluation will include the completeness and clarity of project documentation, including setup instructions, user guides, API documentation, and code comments, facilitating understanding and collaboration among developers.

B. END SEMESTER EXAMINATION (Practical & Oral Exam) (50 Marks)

a) Task Execution: 25 Marks
   Students will be given task (different task for every student) to execute and will be evaluated as per the parameters mentioned in continuous evaluation
b) Mini Project Presentation: 25 Marks
   Students will give mini project presentation and will be evaluated as per the parameters mentioned in continuous evaluation

Two examiners, one Internal and one External will do the evaluation.

Course Type	Course Code	Course Name	Credits
MNP	CSMNP402	MINI PROJECT- 1B	01

Examination Scheme		
Continuous Assessment	**Practical /Oral**	**Total**
50	50	100

Program Outcomes addressed:

1. PO1 : Engineering knowledge
2. PO2 : Problem Analysis
3. PO3 : Design/Development of Solutions
4. PO4 : Conduct investigations of complex problems
5. PO5 : Engineering Tool Usage
6. PO6 : The Engineer & The World
7. PO7 : Ethics
8. PO8 : Individual & Collaborative team work
9. PO9 : Communication
10. PO10 : Project Management & Finance
11. PO11 : Life-long learning

Course Objectives:

1. To familiarize students about available infrastructure at Department/Institute level, online resources, plagiarism, expectations from MP 1A and 1B, etc.
2. To guide students in identifying societal or research needs and formulating them into problem statements.
3. To facilitate problem-solving in group settings.
4. To apply basic engineering principles to address identified problems.
5. To foster self-learning and research skills.

Guidelines for the Mini Project
• At the beginning of semester-III, project guides are required to conduct around 4 hours' orientation sessions including following topics: ➢ Familiarizing students about infrastructure available at Department/Institute level and how to use it. ➢ How to identify societal problems and formulate project problem statement. ➢ How to carry out literature survey.

- What is plagiarism and what care needs to be taken while writing a report.
- What is project report template and how it should be used.
- What are expectations from mini-projects 1A and 1B.

• Mini project may be carried out in one or more form of following:
Product preparations, prototype development model, fabrication of set-ups, laboratory experiment development, process modification/development, simulation, software development, integration of software (frontend-backend) and hardware, statistical data analysis, creating awareness in society/environment etc.

• Students must form groups of 3 to 4 members either from the same or from different departments.

• Groups should conduct surveys to identify needs and develop problem statements in consultation with faculty.

• An implementation plan in Gantt/PERT/CPM chart format covering weekly activities must be submitted.

• Each group must maintain a logbook to record weekly progress, to be verified by the faculty supervisor.

• Faculty input should emphasize guiding by faculty and self-learning by group members.

• Groups should propose multiple solutions, select the best one in consultation with the supervisor, and develop a working model.

• The solution to be validated with proper justification and report to be compiled in standard format of the Institute. Research papers, competition certificates may be submitted as part of annexure to the report.

• With the focus on self-learning, innovation, addressing societal/research/innovation problems and entrepreneurship quality development within the students through the Mini Projects, it is preferable that a single project of appropriate level and quality be carried out in two semesters by all the groups of the students.

• However, based on the individual students or group capability, with the mentor's recommendations, if the proposed Mini Project adhering to the qualitative aspects mentioned above, gets completed in odd semester, then that group can be allowed to work on the extension of the Mini Project with suitable improvements/modifications or a completely new project idea in even semester. This policy can be adopted on a case by case basis.

Course Outcomes:

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

1. Identify problems based on societal or research needs and methodology for solving them.
2. Apply knowledge and skills to solve societal problems collaboratively.
3. Develop interpersonal skills necessary for teamwork.
4. Analyze, verify, and validate results effectively through various methodologies, including, test cases/benchmark data/theoretical/inferences/experiments/simulations, etc.
5. Evaluate the societal and environmental impacts of proposed solutions.
6. Adhere to standard engineering practices.
7. Excel in written and oral communication by technical report writing, oral presentation, and publishing results in
    - Research/white paper/article/blog writing/publication, etc.
    - Business plan for entrepreneurship product creation
    - Patent filing/copyright.
8. Gain technical competencies by participating in competitions, hackathons, etc.
9. Demonstrate lifelong learning capabilities through self-directed group projects.
10. Apply project management principles effectively.

In-Semester Continuous Assessment and End-Semester Examination Guidelines:

- The Head of the Departments will assign a guide to each of the mini-projects and shall form a progress monitoring committee. The guide will carry out weekly monitoring of the project's progress. The committee shall carry out in-semester project evaluation based on presentations with a minimum of two evaluations per semester.
- Assessment will be based on individual contributions, understanding, and responses to questions asked.
- Continuous Assessment marks distribution in semester IV (50 marks):
    - 15 marks for the In-Semester Two Presentations
    - 05 marks for the Participation in Project Competitions, TPP, etc.
    - 25 marks for the Final Report & Presentation
    - 05 marks for Regularity and active participation

The review/progress monitoring committee will assess projects based on the following criteria.

Semester IV:

- Expected tasks include procuring components/systems, constructing a working prototype, and validating results based on prior semester work.
- Reviews will be conducted as follows:

- The first review will assess the readiness to build a working prototype.
- The second review will involve a poster presentation and demonstration of the working model in the last month of the semester.

In addition to the above-mentioned points, the following performance criteria shall be included during the in-semester continuous assessment:

1. Quality of survey and need identification.
2. Clarity and innovativeness in problem definition and solutions.
3. Requirement gathering feasibility study, cost-effectiveness, and societal impact of proposed solutions.
4. Completeness and full functioning of the working model.
5. Effective use of skill sets and engineering norms.
6. Verification & validation of the solutions/test cases.
7. Individual contributions to the group.
8. Clarity in written and oral communication.
9. Participation in technical paper presentations/project competitions/hackathon competitions, etc.

End-Semester Examination in Semester IV (50 marks):

1. Presentation and demonstration to internal and external examiners: 20 marks.
2. Emphasis on problem clarity, innovativeness, societal impact, functioning of the model, skill utilization, and communication clarity: 30 marks.

Course Type	Course Code	Course Name	Credits
VEC	VEC402	ENVIRONMENT & SUSTAINABILITY	02

Examination Scheme		
Continuous Assessment	**End Semester Examination**	**Total**
50	-	50

Program Outcomes addressed:

1. PO2: Problem Analysis
2. PO6: The Engineer & The world
3. PO7: Ethics
4. PO11: Life-long learning

Course Objectives:

1. To provide students with foundational knowledge and understanding of environmental science principles and concepts.
2. To explore the principles of sustainability and their applications in various domains of engineering and technology.
3. To familiarize students with the legal and ethical considerations associated with environmental management and sustainability practices.
4. To equip students with practical skills and strategies for promoting renewable energy, energy efficiency, waste management, and environmental impact assessment.

Module	Details
01.	**Foundations of Environmental Sciences** Introduction to Environmental Science, Earth's Systems: Atmosphere, Hydrosphere, Lithosphere, Biosphere, Ecological Principles: Energy flow, Nutrient cycling, Biodiversity, Environmental Degradation: Pollution, Deforestation, Habitat loss, Environmental Monitoring and Data Analysis.
02.	**Sustainability Basics** Concepts of Sustainability and Sustainable Development, Sustainable Resource Management: Water, Air, Land, Sustainable Agriculture and Food Systems, Sustainable Transportation and Urban Planning, Sustainable Business Practices and Corporate Social Responsibility
03.	**Legal & Ethical Considerations** Environmental Laws and Regulations: National and International Perspectives, Environmental Policies and Governance Frameworks, Ethical Issues in Environmental Decision Making, Environmental Justice and Equity, Corporate Ethics and Environmental Responsibility
04.	**Renewable energy & Energy efficiency**

	Introduction to Renewable Energy Sources: Solar, Wind, Hydro, Biomass, Geothermal, Energy Conversion Technologies and Systems Energy Efficiency Measures and Strategies, Policy Support for Renewable Energy Deployment, Economic and Environmental Impacts of Renewable Energy
05.	**Waste management & recycling** Solid Waste Management: Collection, Treatment, Disposal, Recycling Processes and Technologies, E-waste Management and Hazardous Waste Handling, Circular Economy Principles, Waste Reduction Strategies: Source Reduction, Reuse, Repair
06.	**Environmental Impact Assessment** Introduction to Environmental Impact Assessment (EIA), EIA Process: Screening, Scoping, Impact Assessment, Mitigation, Monitoring, Methods and Tools for Impact Assessment: GIS, LCA, Risk Assessment, Case Studies of EIA in Various Sectors: Infrastructure, Energy, Mining, Construction, Role of Stakeholders in EIA Process
Total no. of hours: **30**	

Course Outcomes:

1. Gain a comprehensive understanding of key environmental science principles and their relevance to engineering disciplines.
2. Apply principles of sustainability to analyze and address environmental challenges in engineering projects and processes.
3. Demonstrate awareness of legal and ethical considerations in environmental decision-making and management practices.
4. Develop proficiency in implementing renewable energy technologies and energy-efficient practices in engineering designs and operations.
5. Acquire knowledge and skills in waste management, recycling, and circular economy principles for sustainable resource utilization.
6. Apply environmental impact assessment methods to evaluate and mitigate the environmental impacts of engineering projects and activities.

Text Books:

1. Environmental Science: Toward a Sustainable Future by Richard T. Wright and Dorothy F. Boorse (Publisher: Pearson Education)
2. Introduction to Environmental Engineering and Science by Gilbert M. Masters and Wendell P. Ela (Publisher: Pearson Education)
3. Renewable and Efficient Electric Power Systems by Gilbert M. Masters (Publisher: Wiley)

Reference Books:

1. Environmental Law Handbook by Thomas F. P. Sullivan, David R. Buente Jr., and Sally Fairfax, Bernan Press
2. Sustainability Science by Bert J. M. de Vries, Springer
3. Environmental Impact Assessment: Theory and Practice by Peter Wathern, Routledge

Other Resources:

1. NPTEL Course: Introduction to Environmetal Engineering & Science- Fundamental & Sustainability Concepts, Prof.Brajesh Kumar Dubey, Department of Multidisciplinary IIT Kharagpur :-Web link https://archive.nptel.ac.in/courses/127/105/127105018/

2. NPTEL Course: Environment And Development, By Prof. Ngamjahao Kipgen, IIT Guwahati, Web link- https://onlinecourses.nptel.ac.in/noc23_hs133/preview