SEMESTER 2

PERIOD: September 2023- December 2023

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY
Department of Information Technology, Programme: Information Technology

Vision

To evolve into a centre of excellence in information technology by creation and exchange of knowledge through leading edge research, innovation and services, which will in turn contribute towards solving complex societal problems and thus building a peaceful and prosperous mankind.

Mission

To impart high quality technical education, research training, professionalism and strong ethical values in the young minds for ensuring their productive careers in industry and academia so as to work with a commitment to the betterment of mankind.

Programme Educational Objectives (PEO)

Graduates of Information Technology program shall

PEO 1: Have strong technical foundation for successful professional careers and to evolve as key-players/ entrepreneurs in the field of information technology.

PEO 2: Excel in analyzing, formulating and solving engineering problems to promote life-long learning, to develop applications, resulting in the betterment of the society.

PEO 3: Have leadership skills and awareness on professional ethics and codes.

Programme Outcomes (PO)

Information Technology Program Students will be able to:

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6.The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7.Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8.Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

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

PO10.Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11.Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSO)

Information Technology Program Students will be able to:

PSO1: Acquire skills to design, analyze and develop algorithms and implement those using high-level programming languages.

PSO2: Contribute their engineering skills in computing and information engineering domains like network design and administration, database design and knowledge engineering.

PSO3: Develop strong skills in systematic planning, developing, testing, implementing and providing IT solutions for different domains which helps in the betterment of life.

Discrete Mathematical Structures

COURSE INFORMATION SHEET

PROGRAMME:**INFORMATION TECHNOLOGY**	DEGREE: **BTECH**
COURSE: **DISCRETE MATHEMATICAL STRUCTURES**	SEMESTER: **III** CREDITS: **4**
COURSE CODE**:** **101903/MA300A** REGULATION: **2021**	COURSE TYPE: **CORE**
COURSE AREA/DOMAIN: **Logic Development**	CONTACT HOURS: **3+1(Tutorial) hours/Week**.
CORRESPONDING LAB COURSE CODE (IF ANY):	LAB COURSE NAME:

SYLLABUS:

UNIT	DETAILS	HOURS
I	**Module- 1 (Fundamentals of Logic)** Mathematical logic: Basic connectives and truth table. Statements –Logical Connectives – Tautology – Contradiction. Logical Equivalence: The Laws of Logic, The principle of duality- Substitution Rules – The implication-The Contrapositive- the Converse – the Inverse. Logical Implication - Rules of Inference, The use of Quantifiers: Open Statement-Quantifier- Logically Equivalent – Contrapositive –Converse –Inverse , Logical equivalences and implications for quantified statement- implications ,negation	9
II	**Module 2:(Fundamentals of Counting Theory)** The Rule of Sum – Extension of Sum Rule - The Rule of Product - Extension of Product Rule – Permutations Combinations, The Binomial Theorem (without proof),Combination with Repetition. The Pigeonhole Principle, The principle of Inclusion and Exclusion Theorem (Without Proof) Generalisation of the principle. Derangements-Nothing in it's right place.	9
	Module 3 (Relations and Functions) Cartesian Product - Binary Relation, Function – domain – range, One to One function, Image restriction, Properties of Relations- Reachability Relations- Reflexive Relations-Symmetric Relations-Transitive relations – Antisymmetric, Relations, Partial Order relations, Equivalence Relation- irreflexive relations. Computer Recognition: Zero-one Matrices - Composite Relations- Zero One Matrix-Relation Matrix. Partially ordered Set –Hasse Diagram-Maximal-Minimal Element-Least upper bound (lub)- Greatest Lower bound(glb) (**From section 7.2, graph theory excluded .From 7.3,Topological sorting Algorithm-excluded**) Equivalence Relations and Partition**s** - Equivalence Class, Lattice- Dual Lattice – Sub lattice – Properties of glb and lub – Properties of Lattice - Special Lattice : Complete Lattice – Bounded Lattice – Completed Lattice – Distributive Lattice	9

IV	**Module- 4 (Generating Functions and Recurrence Relations)** Generating Function -First order linear recurrence relations with constant Coefficients- homogeneous nonhomogeneous- Solution. The second-order linear recurrence relation	9
	with Constant Coefficients - homogeneous- Nonhomogeneous – Solution.- Three cases, The Nonhomogeneous Recurrence Relation -First order- Second Order	
V	**Module-5 (Algebraic Structures)** Algebraic system-Properties- Homomorphism and Isomorphism. Semigroup and monoid– cyclicmonoid, sub semigroup and sub monoid- Homomorphism and Isomorphism of Semigroup and monoids. Grou**p-** Elementary properties- subgroup- symmetric group on three symbols-The direct product of two groups. Group Homomorphism- Isomorphism of groups, Cyclic group. Right coset left cosets- Lagrange's Theorem	9
	TOTAL HOURS	45

TEXT/REFERENCE BOOKS:

Text Book

1. Discrete and Combinatorial Mathematics (An Applied Introduction), Ralph P Grimaldi, B VRamana , 5th Edition, Pearson

Reference Books

1) Kenneth H. Rosen, Discrete Mathematics and Its Applications with Combinatorics and GraphTheory, Seventh Edition, MGH, 2011
2) Trembly J.P and Manohar R, "Discrete Mathematical Structures with Applications toComputer Science", Tata McGrawHill Pub. Co. Ltd., New Delhi, 2003.
3) Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, "Discrete MathematicalStructures", Pearson Education Pvt Ltd., New Delhi, 2003
4) Kenneth H.Rosen, "Discrete Mathematics and its Applications", 5/e, Tata Mc Graw Hill

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
	Maths they studied at school level		

COURSE OBJECTIVES:

1	To introduce mathematical notations and concepts in discrete mathematics that is essential for computing
2	To train on mathematical reasoning and proof strategies.
3	To cultivate analytical thinking and creative problem solving skills

COURSE OUTCOMES:

SL.NO	DESCRIPTION	Blooms' Taxonomy Level
C203.1	Students will be able to Learn the fundamentals of enumeration orcounting techniques and methods of arrangements and derangements.	Apply (level 3)
C203.2	Students will be able to Learn the fundamentals of propositional logic and predicate calculus and apply to test the validity of statements	Validate (level 4)
C203.3	Students will be able to Learn the ideas of relations, functions equivalence relation and posets and it's applications	Construct (level 5)
C203.4	Students will Understand recurrence relation and apply the method of solving different type of recurrence relations using generating functions	Solve/Apply (level 3)
C203.5	Students will be able to Understand Fundamentals of Algebraic structuresits properties such as monoids and groups	Solve/Apply (level 3)

CO-PO AND CO-PSO MAPPING

	PO1	PO2	PO3	PO 4	PO5	PO 6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C203.1	3	2	3	2	1	_	_	_	_	2	_	1	2	_	_
C203.2	3	2	3	2	3	_	_	_	_	2	_	2	2	_	_
C203.3	3	2	2	_	2	_	_	_	_	2	_	2	-	_	_
C203.4	3	2	3	1		_	_	_	_	2	_	_	_	_	_
C203.5	3	2	2	1	1	_	_	_	_	2	_	1	-	_	_

JUSTIFICATIONS FOR THE MAPPING

Mapping	LOW/MEDIUM/HIGH	Justification
CS203.1-PO1	H	The arrangement and combinations of data to be taken for different problems can be identified
CS203.1-PO2	M	Counting techniques can be used to reach conclusions in the problems involving huge data
CS203.1-PO3	H	Counting techniques will help life-long learning in the broadest context of technological change.
CS203.1-PO4	M	Counting techniques can be Used research-based knowledge and research methods.
CS203.1-PO5	L	Used to select, and apply appropriate techniques with an understanding of the limitations.
CS203.1-PO10	M	Knowing Counting techniques will help in Communicating effectively on complex engineering activities with the engineering community and with society
CS203.1-PO12	L	Techniques will help life-long learning in the broadest context of technological change.
CS203.2-PO1	H	The validity of facts can be verified using predicate and propositional logic
CS203.2-PO2	M	The real life events can be represented and verified using Mathematical logic
CS203.2-PO3	H	Reasoning is made possible for engineering problems
CS203.2-PO4	M	Logic Uses research-based knowledge and research methods including design of experiments
CS203.2-PO5	L	Knowledge about logic can Create, select, and apply appropriate techniques with an understanding of the limitations.
CS203.2-PO10	M	Knowing logic will help in Communicating effectively on complex engineering activities with the engineering community and with society
CS203.2-PO12	M	Logic will help life-long learning in the broadest context of technological change.
CS203.3 -PO1	H	The reasoning and inferences made by them can be substantiated by the various proof techniques
CS203.3-PO2	M	The proof techniques can be used to verify the complex engineering solutions
CS203.3-PO5	M	The arrangement and combinations of data to be taken for different problems can be identified
CS203.3-PO10	M	discrete structures can be used to reach conclusions in the problems involving huge data

CS203.3-PO12	M	discrete structures will help life-long learning in the broadest context of technological change.
CS203.4-PO1	H	The concepts of discrete structures can be used to solve various complex engineering problems
CS203.4-PO2	M	The knowledge about the discrete computational structures will help them to reach conclusions about the complexity and methodologies for solving real life problems
CS203.4-PO3	H	Discrete structures can aid in the representation of various real life Problems
CS203.4-PO4	L	The knowledge about the discrete computational structures Use research-based knowledge and research methods including design of experiments
CS203.4-PO10	M	Knowing Discrete structures will help to Communicate effectively on complex engineering activities with the engineering community and with society
CS203.5-PO1	H	Algebraic structures can be used to visualize the complex engineering problems involving sets of data
CS203.5-PO2	M	The similarity and characteristics of data can be analyzed using algebraic principles
CS203.5-PO3	M	It can be used to compare and contrast the complexity of algorithms that were developed
CS203.5-PO4	L	It helps to analyze the complexity and choose the best method for the particular problem
CS203.5-PO5	L	All algebraic structures can be compared using a single measure to identify the amount of computations involved in them so that the optimal one can be identified
CS203.5-PO10	M	The proof techniques can be used to verify the complexengineering solutions
CS203.5-PO12	L	Algebraic structures will help life-long learning in the broadest contextof technological change

GAPS IN THE SYLLABUS - TO MEET INDUSTRY/PROFESSION REQUIREMENTS:

Si NO	DESCRIPTION	PROPOSED ACTIONS	RELEVANCE WITH POs	RELEVANCE WITH PSOs
1	Graph Theory and its applications	Seminar	1	1
2	Plotting graph for a function	Class lecturing along with the topic of function	1	1

PROPOSED ACTIONS: TOPICS BEYOND SYLLABUS/ASSIGNMENT/INDUSTRY VISIT/GUEST LECTURER/NPTEL ETC

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

Si NO	DESCRIPTION	PROPOSED ACTIONS	RELEVANCE WITH POs	RELEVANCE WITH PSOs
1	Graph Theory	Class Assignment	1,3	1

WEB SOURCE REFERENCES:

1	http://web.stanford.edu/class/cs103x/cs103x-notes.pdf
2	https://www.tutorialspoint.com/discrete_mathematics/discrete_mathematics_recurrence_relation.htm
3	http://nms.lu.lv/wp-content/uploads/2016/04/21-linear-recurrences.pdf
4	http://wwwhome.cs.utwente.nl/~infrieks/MHMI/2005.jk.pdf
5	http://nicolas.thiery.name/macs358/Notes/AlgebraicStructures.pdf

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

✓ CHALK & TALK	✓ STUD. ASSIGNMENT	✓ WEB RESOURCES	
✓ LCD/SMART BOARDS	✓ STUD. SEMINARS	☐ ADD-ON COURSES	

ASSESSMENT METHODOLOGIES-DIRECT

✓ ASSIGNMENTS	✓ STUD. SEMINARS	✓ TESTS/MODEL EXAMS	✓ UNIV. EXAMINATION

☐*STUD. LAB PRACTICES*	☐STUD. VIVA	☐MINI/MAJOR PROJECTS	☐CERTIFICATIONS
☐*ADD-ON COURSES*	☐OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

☑*ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)*	☑STUDENT FEEDBACK ON FACULTY (TWICE)
☐*ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS*	☐OTHERS

Prepared by

Menny M.N

Approved by

Dr. Binu.R

101903/MA300A Discrete Mathematical

StructuresCourse Content and Lecture Schedule

No	Contents	No. of Lecture Hrs
Module – 1 (Fundamentals of Logic) (9 Hrs.)		
1.1	Mathematical logic, Basic Connectives and Truth Table	1
1.2	Statements, Logical Connectives, Tautology, Contradiction	1
1.3	Logical Equivalence, The Laws of Logic	1
1.4	The Principle of duality, Substitution Rules	1
1.5	The implication, The Contrapositive, the Converse , the Inverse	1
1.6	Logical Implication, Rules of Inference, Logical Implication	1
1.7	The use of Quantifiers, Open Statement, Quantifier, Negation	1
1.8	Logically Equivalent, Contrapositive, The Converse, The Inverse	1
1.9	Logical Implications	1
Module - 2 (Fundamentals of Counting Theory) (9 Hrs)		
2.1	The Pigeon-hole Principle	1
2.2	The Rule of Sum	1
2.3	Extension of Sum Rule	1
2.4	The Rule of Product	1
2.5	Extension of Product Rule , Permutations	1
2.6	Combinations, Combination with repetition	2
2.7	The Binomial Theorem	1
2.8	The Principle of Inclusion and Exclusion Theorem (Without Proof)	1
Module - 3 (Relations and Functions) (9 Hrs)		
3.1	Cartesian Product, Binary Relation, Function, Domain, Range , One to One Function Image	1

3.2	Properties, Reflexive Relations, Symmetric Relations, Transitive relations, Antisymmetric Relations.	1
3.3	Partial Order relations	1
3.4	Equivalence Relation, Irreflexive Relations.	1
3.5	Partially ordered Set, Hasse Diagram.	1
3.6	Maximal-Minimal Element, Least Upper bound, Greatest Lower Bound	1
3.7	Equivalence Relations and Partitions ,Equivalence Class	1
3.8	Lattice- Dual Lattice,sub lattice , Properties of glb and lub	1
3.9	Properties of Lattice , Special Lattice , Complete Lattice, Bounded Lattice, Completed Lattice, Distributive Lattice	1
Module - 4 (Generating Functions and Recurrence Relations) (9 Hrs)		
4.1	Generating Function , Definition and Examples	1
4.2	Exponential Generating Function.	1
4.3	First Order Linear Recurrence Relations with Constant Coefficients	1
4.4	Homogeneous Solution	1
4.5	Non homogeneous Solution	1
4.6	Second order linear recurrence relations with constant coefficients	2
4.7	Homogeneous Solution	1
4.8	Non homogeneous Solution	1
Module - 5 (Algebraic Structures)(9 Hrs)		
5.1	Algebraic System-Properties, Homomorphism and Isomorphism	1
5.2	Semi group , Monoid, Cyclic monoid	1
5.3	Sub semigroup and sub monoid	1
5.4	Homomorphism and Isomorphism of Semigroup, Monoids and Groups	1
5.5	Elementary Properties, Subgroup, Symmetric group on three symbols	1

5.6	The direct Product of two Groups	1
5.7	Group Homomorphism, Isomorphism, Cyclic group	1
5.8	Right coset, Left coset	1
5.9	Lagrange's Theorem	1

DATA STRUCTURES

COURSE INFORMATION SHEET

PROGRAMME: **Computer Science & Engineering/Information Technology/Artificial Intelligence**	**DEGREE: BTECH**
COURSE: **DATA STRUCTURES**	SEMESTER: III CREDITS: 4
COURSE CODE: 101903/C0300B REGULATION: 2021	COURSE TYPE: **CORE**
COURSE AREA/DOMAIN: PROGRAMMING	CONTACT HOURS: **3+1(Tutorial) hours/Week.**
CORRESPONDING LAB COURSE CODE (IF ANY): 101903/CO322S	LAB COURSE NAME: DATA STRUCTURES LAB

SYLLABUS:

UNIT	DETAILS	HOURS
I	**Basic Concepts of Data Structures** System Life Cycle, Algorithms, Performance Analysis, Space Complexity, Time Complexity, Asymptotic Notation, Complexity Calculation of Simple Algorithms	5
II	**Arrays and Searching** Polynomial representation using Arrays, Sparse matrix, Stacks, Queues-Circular Queues, Priority Queues, Double Ended Queues, Evaluation of Expressions, Linear Search and Binary Search	10
III	**Linked List and Memory Management** Self Referential Structures, Dynamic Memory Allocation, Singly Linked List-Operations on Linked List. Doubly Linked List, Circular Linked List, Stacks and Queues using Linked List, Polynomial representation using Linked List	12
IV	Trees and Graphs Trees, Binary Trees-Tree Operations, Binary Tree Representation, Tree Traversals, Binary Search Trees- Binary Search Tree Operations Graphs, Representation of Graphs, Depth First Search and Breadth First Search on Graphs, Applications of Graphs	8
V	**Sorting and Hashing** Sorting Techniques – Selection Sort, Insertion Sort, Quick Sort, Merge Sort and Heap Sort Hashing- Hashing Techniques, Collision Resolution, Overflow handling, Hashing functions –Mid square, Division, Folding, Digit Analysis	10
	TOTAL HOURS	45

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T/R	Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Universities Press, Fundamentals of Data Structures in C
R	Samanta D., Classic Data Structures, Prentice Hall India.
R	Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, 2/e, Cengage Learning.
R	Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication.
R	Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill.
R	Peter Brass, Advanced Data Structures, Cambridge University Press.
R	Lipschuts S., Theory and Problems of Data Structures, Schaum's Series.
R	Wirth N., Algorithms + Data Structures = Programs, Prentice Hall.
R	Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI.
R	Martin Barrett, Clifford Wagner, C And Unix: Tools For Software Design, John Wiley

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
101903/CO200G	Programming in C	Fundamentals of C programming	S2

COURSE OBJECTIVES:

1	To mould the learner to understand the various data structures, their organization and operations.
2	To help the learners to assess the applicability of different data structures and associated algorithms for solving real world problem which requires to compare and select appropriate data structures to solve the problem efficiently.

3	To introduce abstract concepts for data organization and manipulation using data structures such as stacks, queues, linked lists, binary trees, heaps and graphs for designing their own data structures to solve practical application problems in various fields of Computer Science

COURSE OUTCOMES:

Students will be able to

101903/CO300B.1	Design an algorithm for a computational task and calculate the time/space complexities of that algorithm	Cognitive Knowledge Level: Apply
101903/CO300B.2	Identify the suitable data structure (array or linked list) to represent a data item required to be processed to solve a given computational problem and write an algorithm to find the solution of the computational problem	Cognitive Knowledge Level: Apply
101903/CO300B.3	Write an algorithm to find the solution of a computational problem by selecting an appropriate data structure (binary tree/graph) to represent a data item to be processed	Cognitive Knowledge Level: Apply
101903/CO300B.4	Store a given dataset using an appropriate Hash Function to enable efficient access of data in the given set	Cognitive Knowledge Level: Apply
101903/CO300B.5	Select appropriate sorting algorithms to be used in specific circumstances	Cognitive Knowledge Level: Analyze
101903/CO300B.6	Design and implement Data Structures for solving real world problems efficiently	Cognitive Knowledge Level: Apply

CO-PO AND CO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
101903/CO300B.1	3	3	2	2	-	1	-	-	-	-	-	2	2	3	1
101903/C	2	2	3	2	-	1	-	-	-	-	-	1	2	2	-

O300B.2															
101903/CO300B.3	3	2	3	2	-	2	-	-	-	-	-	1	1	2	-
101903/CO300B.4	3	1	2	3	-	1	-	-	-	-	-	2	2	2	2
101903/CO300B.5	3	2	3	3	-	1	-	-	-	-	-	2	1	2	-
101903/CO300B.6	3	2	3	2	-	2	-	-	-	-	-	2	3	3	-
101903/CO300B	**2**	**2**	**2**	**2**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**2**	**1**

1-LOW, 2-MEDIUM, 3-HIGH

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	LOW/MEDIUM/HIGH	Justification
101903/CO300B.1-PO1	HIGH	The knowledge in designing an algorithm for a task and calculating the time space complexities of the algorithm helps in designing solutions for complex engineering problems.
101903/CO300B.1-PO2	HIGH	The knowledge of analyzing the space and time complexities of the algorithm helps in analyzing complex engineering problems and reach substantiated conclusion.
101903/CO300B.1-PO3	MEDIUM	The design of solutions for complex problems requires the knowledge to write an efficient algorithm and also to calculate the time and space complexities.
101903/CO300B.1-PO4	MEDIUM	Conducting investigation of complex problems to reach valid conclusions require the skill of designing an efficient algorithm.
101903/CO300B.1-PO6	LOW	The knowledge of calculating time and space complexities of the algorithm helps to design an efficient algorithm and thereby an efficient product useful to the society.
101903/CO300B.1-PO12	MEDIUM	The technological changes that happen every day require the skill of writing / designing efficient

		algorithms and hence can be considered as a lifelong learning.
101903/CO300B.1-PSO1	MEDIUM	The knowledge in designing algorithms and asymptotic notations help in designing solutions and analyzing its complexity.
101903/CO300B.1-PSO2	HIGH	The ability to design an efficient algorithm and calculate its complexity helps to enhance the programming skills and develop efficient programs
101903/CO300B.1-PSO3	LOW	These concepts of designing algorithms and calculating space and time complexities are fundamental to CS and can be used in research and other innovative ideas.
101903/CO300B.2-PO1	MEDIUM	The knowledge of arrays, linked lists, stacks and queues can be applied to solve complex engineering problems.
101903/CO300B.2-PO2	MEDIUM	The knowledge of arrays or linked list helps to use them appropriately in programs and observe the output of it thereby reaching substantiated conclusion.
101903/CO300B.2-PO3	HIGH	The knowledge of arrays, linked lists, stacks and queues can be applied to design solutions to complex engineering problems.
101903/CO300B.2-PO4	MEDIUM	The ability to use the data structure arrays, linked lists, stacks and queues help us to implement efficient algorithms, code efficient programs and thereby perform analysis and interpretation of data to reach valid conclusions.
101903/CO300B.2-PO6	LOW	The use of arrays, linked lists help to design efficient algorithms useful to the society.
101903/CO300B.2-PO12	LOW	The knowledge of using linked lists and arrays and appropriately choosing them based on the application helps to easily adapt to technological changes and thereby conduct lifelong learning.
101903/CO300B.2-PSO1	MEDIUM	The knowledge of arrays, linked lists, stacks and queues can be applied to design solutions to complex engineering problems in multidisciplinary areas. They belong to the core concepts of CS.
101903/CO300B.2-PSO2	MEDIUM	The knowledge of arrays, linked lists, stacks and queues can be applied to design solutions and thereby design good and efficient programs . This enhances the

		software skills of the students.
101903/CO300B.3-PO1	HIGH	The knowledge of finding the solution of a computational problem by selecting non linear data structures like binary trees or graph can be applied to solve complex engineering problems.
101903/CO300B.3-PO2	MEDIUM	The knowledge of finding the solution of a computational problem by selecting non linear data structures like binary trees or graph helps to use them appropriately in programs and observe the output of it thereby reaching substantiated conclusion.
101903/CO300B.3-PO3	HIGH	The knowledge of finding the solution of a computational problem by selecting non linear data structure like binary trees or graph can be applied to design solutions to complex engineering problems.
101903/CO300B.3-PO4	MEDIUM	The ability to use the data structure like binary trees and graphs help us to implement efficient algorithms, code efficient programs and thereby perform analysis and interpretation of data to reach valid conclusions.
101903/CO300B.3-PO6	MEDIUM	The use of non linear data structures like binary trees and graphs helps the engineers design efficient algorithms useful to the society like storing huge databases and finding the shortest distance using graph algorithms
101903/CO300B.3-PO12	LOW	The knowledge of using linked lists and arrays and appropriately choosing them based on the application helps to easily adapt to technological changes and thereby conduct lifelong learning.
101903/CO300B.3-PSO1	LOW	The knowledge of non linear data structures like trees and graphs can be applied to design solutions to complex engineering problems.
101903/CO300B.3-PSO2	MEDIUM	This knowledge helps in designing efficient algorithms using appropriate data structure.
101903/CO300B.4-PO1	HIGH	The knowledge of storing and accessing data efficiently by using appropriate Hash functions plays a role in designing solutions to complex engineering problems.
101903/CO300B.4-PO2	LOW	The knowledge of storing and accessing data efficiently by using appropriate Hash functions helps to formulate

		an efficient solution to complex engineering problems
101903/CO300B.4-PO3	MEDIUM	The knowledge of storing and accessing data efficiently by using appropriate Hash functions helps to design solutions to complex problems thereby helping the society.
101903/CO300B.4-PO4	MEDIUM	The knowledge of storing and accessing data efficiently by using appropriate Hash functions helps to store data for conducting experiments or analysis.
101903/CO300B.4-PO6	LOW	The knowledge of storing and accessing data efficiently by using appropriate Hash functions helps to store data and use it for engineering practices in an efficient manner.
101903/CO300B.4-PO12	MEDIUM	The knowledge of storing and accessing data efficiently by using appropriate Hash functions helps to easily adapt to technological changes and thereby conduct lifelong learning
101903/CO300B.4-PSO1	MEDIUM	This basic knowledge of storing and accessing data efficiently by using appropriate Hash functions can be used in designing solutions to complex multidisciplinary engineering problems.
101903/CO300B.4-PSO2	MEDIUM	This basic knowledge of storing and accessing data efficiently by using appropriate Hash functions is needed while designing solutions and thereby delivering quality products meeting the industry needs.
101903/CO300B.4-PSO3	MEDIUM	This basic knowledge of storing and accessing data efficiently by using appropriate Hash functions are fundamental to the CS discipline and can be used in designing algorithms for complex problems.
101903/CO300B.5-PO1	HIGH	The ability to select appropriate sorting algorithms helps in solving complex engineering problems.
101903/CO300B.5-PO2	MEDIUM	The ability to select appropriate sorting algorithms and use it in specific environment helps to analyze various parameters needed to decide the efficiency of the solution to a complex problem.
101903/CO300B.5-PO3	HIGH	The ability to select appropriate sorting algorithms helps to design solutions to complex engineering problems in an efficient manner.

101903/CO300B.5-PO4	HIGH	The ability to select appropriate sorting algorithms and use it in specific environments helps to investigate complex problems and reach valid conclusions.
101903/CO300B.5-PO6	LOW	The ability to select appropriate sorting algorithms helps to design and develop quality products that helps the society.
101903/CO300B.5-PO12	MEDIUM	The ability to select appropriate sorting algorithms helps to easily adapt to technological changes and thereby conduct lifelong learning
101903/CO300B.5-PSO1	LOW	The ability to select and use appropriate sorting algorithms is a computer science specific skill needed for designing solutions for complex engineering problems.
101903/CO300B.5-PSO2	MEDIUM	The ability to select and use appropriate sorting algorithms helps to develop efficient algorithms and thereby solve complex problems.
101903/CO300B.6-PO1	HIGH	The ability to design and implement data structures for solving real world problems efficiently is a specialization acquired by the programmers that enables them to develop efficient solutions to complex engineering problems.
101903/CO300B.6-PO2	MEDIUM	The ability to design and implement data structures for solving real world problems efficiently helps to formulate and analyze complex problems and reach substantiated conclusions.
101903/CO300B.6-PO3	HIGH	The ability to design and implement data structures for solving real world problems efficiently helps to design solutions to complex engineering problems that helps the society.
101903/CO300B.6-PO4	HIGH	The ability to design and implement data structures for solving real world problems efficiently helps to design solutions, analyze and interpret data efficiently and provide valid conclusions.
101903/CO300B.6-PO6	LOW	The ability to design and implement data structures for solving real world problems efficiently helps to develop quality and efficient products helpful to the society.
101903/CO300B.	MEDIUM	The ability to design and implement data structures for

6-PO12		solving real world problems efficiently helps to easily adapt to technological changes and thereby conduct lifelong learning
101903/CO300B.6-PSO1	HIGH	The ability to design and implement data structures for solving real world problems efficiently can be applied in designing solutions to complex multidisciplinary engineering problems.
101903/CO300B.6-PSO2	HIGH	The ability to design and implement data structures for solving real world problems efficiently can be applied in designing algorithms for complex engineering problems and thereby develop quality and efficient products that meet industry standards

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

Sl.No	Description	PO mapping	Proposed Actions
1	AVL Trees	PO1,PO3,PSO2	Learning Materials provided

WEB SOURCE REFERENCES:

1	http://nptel.ac.in/courses/106103069
2	https://ocw.mit.edu/courses/electrical-engineering-and-computer science/6-006-introduction-to-algorithms-spring-2008/
3	http://web.stanford.edu/class/cs166/

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

✓ CHALK & TALK	✓ STUD. ASSIGNMENT	✓ WEB RESOURCES	
✓ LCD/SMART BOARDS	✓ STUD. SEMINARS	☐ ADD-ON COURSES	

ASSESSMENT METHODOLOGIES-DIRECT

✓ ASSIGNMENTS	☐ STUD. SEMINARS	✓ TESTS/MODEL EXAMS	✓ UNIV. EXAMINATION

✓ STUD. LAB PRACTICES	✓ STUD. VIVA	☐ MINI/MAJOR PROJECTS	☐ CERTIFICATIONS
☐ ADD-ON COURSES	☐ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

✓ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	✓ STUDENT FEEDBACK ON FACULTY (ONCE)
☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS	☐ OTHERS

Prepared by **Approved by**

Tinku Soman Jacob **HOD**

Course Plan

No	Topic	No. of Lectures
1	**Module1: Basic Concepts of Data Structures**	
1.1	System Life Cycle,	1 hour
1.2	Algorithms, Performance Analysis	1 hour
1.3	Space Complexity, Time Complexity	1 hour
1.4	Asymptotic Notation (Big O Notation)	1 hour
1.5	Complexity Calculation of Simple Algorithms	1 hour
2	**Module 2: Arrays and Searching**	
2.1	Polynomial representation using Arrays	1 hour
2.2	Sparse matrix	2 hours
2.3	Stacks	1 hour
2.4	Infix to Postfix Conversion	1 hour
2.5	Evaluation of Postfix Expression	1 hour
2.6	Queues, Circular Queues	1 hour
2.7	Priority Queues	1 hour
2.8	Double Ended Queues,	1 hour
2.9	Linear Search and Binary Search	1 hour
3	**Module 3: Linked List and Memory Management**	
3.1	Self-Referential Structures	1 hour
3.2	Dynamic Memory Allocation	1 hour
3.3	Singly Linked List-Operations on Linked List	2 hours
3.4	Doubly Linked List	1 hour
3.5	Circular Linked List	1 hour
3.6	Stacks using Linked List	1 hour
3.7	Queues using Linked List	1 hour

3.8	Polynomial representation using Linked List	2 hours
4	**Module 4: Trees and Graphs**	
4.1	Trees, Binary Trees	1 hour
4.2	Tree Operations, Binary Tree Representation	1 hour
4.3	Tree Traversals	1 hour
4.4	Binary Search Trees	1 hour
4.5	Binary Search Tree Operations	1 hour
4.6	Graphs, Representation of Graphs	1 hour
4.7	Depth First Search and Breadth First Search on Graphs	1 hour
4.8	Applications of Graphs	1 hour
5	**Module 5: Sorting and Hashing**	
5.1	Sorting Techniques – Selection Sort	1 hour
5.2	Insertion Sort	1 hour
5.3	Quick Sort	1 hour
5.4	Merge Sort	1 hour
5.5	Heap Sort	1 hour
5.6	Hashing- Hashing Techniques	1 hour
5.7	Hashing functions – Mid square and Division methods	1 hour
5.8	Folding and Digit Analysis methods	1 hour
5.9	Collision Resolution	1 hour
5.10	Overflow handling	1 hour

Tutorial Questions

1. Write the pseudo code for expression evaluation and identify the worst case time and space complexity.
2. Write the pseudo code for linear search and binary search and identify the worst case time and space complexity.
3. Write the c program code to reverse a string using linked stack.
4. Pseudo-code to merge two sorted singly linked lists into a single sorted linked list.
5. Implement two linked list L1 and L2 and find the following

   a. L1U L2

   b. L1L2

Assignment Questions

Assignment 1

1. Find out the time complexity of

a. biggest of an array of n numbers

b. matrix multiplication

Assignment 2

1.Write an algorithm to check whether the brackets are closed properly?

eg: (2+3)-4) not properly closed.

2.Write an algorithm to reverse a string using stack.

3. Write a program/pseudocode for addition of polynomials containing two variables using linked list.

4. Implement/pseudocode a Doubly Linked List from a string taking each character from the string. Check if the given string is palindrome in an efficient method.

5. The Indian Airline company maintains six scheduled flights per day which they identify by the no's one to six. For each of these flights they keep list of passengers in alphabetical order. Write programs/pseudocode that set up and maintain this database continuously online with add, delete and list facility.

DIGITAL SYSTEM DESIGN

COURSE INFORMATION SHEET

PROGRAMME: INFORMATION TECHNOLOGY	**DEGREE:** BTECH
COURSE: DIGITAL SYSTEM DESIGN	**SEMESTER:** III **CREDITS:** 4
COURSE CODE: 101004/IT300C **REGULATION**: 2020	**COURSE TYPE:** CORE
COURSE AREA/DOMAIN: COMPUTER HARDWARE	**CONTACT HOURS:** 3+1 (Tutorial) hours/Week.
CORRESPONDING LAB COURSE CODE (IFANY): Nil	**LAB COURSE NAME:**Nil

SYLLABUS

Module 1: NUMBER SYSTEM (9 Hours)
Number Systems – Decimal, Binary, Octal, Hexadecimal - conversion from one system to another – Representation of negative numbers using 2's **complement**. Arithmetic Operations – Addition, Subtraction, Multiplication, Division of Binary numbers, **Booth's** algorithm for multiplication, Representation of negative numbers, Representation of floating point numbers. Representation of BCD numbers, BCD Addition, Binary Codes – Gray codes – excess 3 code- Character Coding Schemes – ASCII, EBCDIC.
Module 2: BOOLEAN ALGEBRA & LOGIC GATES (9 Hours)
Boolean Algebra - Postulates of Boolean algebra - Canonical and Standard Forms -Simplification of Boolean Functions using Karnaugh Map - Product-of-Sums Simplification — Don't-Care Conditions – Quine –McClusky minimization technique – Basic Gates-UniversalGates.
Module 3: COMBINATIONAL LOGIC (9 Hours)
Combinational Circuits – Analysis and Design Procedures - Binary **Adder-Subtractor** (Half & Full) - Carry look ahead adder, BCD adder, code converter, - Magnitude Comparator - Decoders – Encoders Parity Generator– Multiplexers – **De-multiplexers** – Implementation of Boolean functions using MUX.
Module 4: SEQUENTIAL LOGIC CIRCUITS (9 Hours)
Sequential Circuits - Storage Elements: Latches , Flip-Flops – RS, JK, D, T, Triggering of flip-flops, Master-Slave- Analysis of Clocked Sequential Circuits - Design Procedure-using JK,D & T.
Module 5: COUNTERS AND SHIFT REGISTERS (9 Hours)
Registers - Shift Registers – SISO, PIPO, SIPO, PISO- Universal shift registers, Counters-Design of Counters- Synchronous & Asynchronous Counters — up-down counter, Decade counter, BCD counter, Johnson counter, Ring counter ,Memory &Programmable logic- RAM, ROM, PLA,PAL

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T	Mano M. M., Digital Logic & Computer Design, 4/e, Pearson Education, 2013.
T	Charles H Roth ,Jr, Lizy Kurian John, Digital System Design using VHDL,2/e, Cengage Learning .
R	Tokheim R. L., Digital Electronics Principles and Applications, 7/e, Tata McGraw Hill, 2007.
R	Mano M. M. and M. D Ciletti, Digital Design, 4/e, Pearson Education, 2008
R	Rajaraman V. and T. Radhakrishnan, An Introduction to Digital Computer Design, 5/e, Prentice Hall India Private Limited, 2012.
R	Leach D, Malvino A P, Saha G, Digital Principles and Applications, 8/e, McGraw Hill Education, 2015.
R	Floyd T. L., Digital Fundamentals, 10/e, Pearson Education, 2009
R	M. Morris Mano, Computer System Architecture, 3/e, Pearson Education, 2007.

COURSE PREREQUISITES:NIL

COURSE OBJECTIVES:

1	To impart an understanding of the basic concepts of Boolean algebra and digital circuitdesign.
2	To provide familiarity with the design and implementation of different types of practicallyused combinational and sequential circuits.
3	To provide an introduction to use Hardware Description Language.

COURSE OUTCOMES:

CO No.	Course Outcome(CO)	Bloom's Category
CO 1	To perform base conversion and arithmetic operations invarious number systems.	Apply
CO 2	To design digital circuits using simplified Boolean functions	Create
CO 3	To develop simple design of combinational circuits	Apply
CO 4	To develop simple design of sequential circuits	Apply
CO 5	To interpret the generalization of synchronous and asynchronous sequential circuits	Understand

CO-PO AND CO-PSO MAPPINOG

CO No.	Programme Outcomes (POs)												Programme-Specific Outcomes (PSOs)		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	3	-	-	3	-	-	-	-	-	-	-	-	-	-	2
2	3	3	3	2	-	-	-	-	-	-	-	-	-	-	2
3	3	3	3	3	-	-	-	-	-	-	-	-	-	-	2
4	3	-	-	3	-	-	-	-	-	-	-	-	-	-	2
5	3	3	-	-	-	-	-	-	-	-	-	-	-	-	2

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
CO1-PO1	H	Knowledge of Boolean algebra helps the students in circuit designing.
CO1-PO4	H	Analysis of logic circuits provides students a better understanding of digital circuits.
CO1-PO12	L	Help the students in the design of simple digital circuits using gates.
C02-PO1	H	Acquire skills to easily simplify Boolean functions.
CO2-PO2	H	Analysis of the combinational circuits to provide simple conclusions.
CO2-PO3	H	Designing of complex combinational circuits is achieved.
CO2-PO4	M	Ability to choose a simplified circuit for implementing a combinational circuit using an appropriate simplification method.
CO2-PO12	M	Knowledge of Boolean algebra helps in the design of efficient circuits.
CO3-PO1	H	Having knowledge in Boolean function, students could develop sequential circuits.
CO3-PO2	H	Knowledge of Flip flops could be used to reduce the complexity of the sequential circuits.

CO3-PO4	H	Having the knowledge in various sequential circuit design principles students could analyze the problem and come to a conclusion on which design principle to be used.
CO4-PO1	H	Choosing the appropriate method to implement the sequential circuit will help in designing efficientcircuits.
CO4-PO4	H	Knowledge of hardware description language to understand the concept of simple circuits.
CO5-PO1	H	Having knowledge of hardware description language helps students to design logic circuits.
CO5-PO2	H	Having knowledge of hardware description language students to analyze complex circuits.
CO5-PO12	M	Students will be able to implement and test the circuits.

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

1	Fault Tolerance

WEB SOURCE REFERENCES:

1	http://nptel.iitm.ac.in/video.php?subjectId=117106086
2	http://www.asic-world.com/digital/tutorial.html
3	http://www.technologystudent.com/elec1/dig1.htm
4	http://cusatbtechguru.blogspot.in/2012/06/eecs-303-digital-electronics-lecture.html

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

☑CHALK & TALK	**☑ STUD. ASSIGNMENT**	**☑WEB RESOURCES**	**☑ LCD/SMART BOARDS**
☑ STUD. SEMINARS	**☐ ADD-ON COURSES**		

ASSESSMENT METHODOLOGIES-DIRECT

☑ ASSIGNMENTS	☑ STUD. SEMINARS	☑ TESTS/MODEL EXAMS	☑ UNIV. EXAMINATION
☐ STUD. LAB PRACTICES	☐ STUD. VIVA	☐ MINI/MAJOR PROJECTS	☐ CERTIFICATIONS

☐ ADD-ON COURSES	☐ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

☑ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	☑ STUDENT FEEDBACK ON FACULTY (TWICE)
☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS	☐ OTHERS

Prepared by
Dr. Ranju S Kartha
(Faculty in Charge)

Approved by
Dr.Neeba E.A
(HOD)

Course Plan

No	Topic	No. of Lectures
1	**Module 1 : Number System**	**9 Hours**
1.1	Number Systems – Decimal – Binary – Octal - Hexadecimal - conversion from one system to another – Representation of negative numbers using 2's compliment.	3 Hours
1.2	Arithmetic Operations – Addition – Subtraction – Multiplication -Division of Binary numbers - Booths algorithm for multiplication - representation of negative numbers - Representation of floatingpoint numbers.	4 Hours
1.3	Representation of BCD numbers - BCD Addition Binary Codes – Gray codes – excess 3 code - Character Coding Schemes – ASCII – EBCDIC.	2 Hours
2	**Module 2 : Boolean Algebra & Logic Gates**	**9 Hours**
2.1	Boolean Algebra - Postulates of Boolean algebra - Canonical and Standard Forms.	2 Hours
2.2	Simplification of Boolean Functions using Karnaugh Map - Product-of-Sums Simplification - Don't-Care Conditions.	2 Hours
2.3	Quine –McClusky minimization technique.	2 Hours
2.4	Basic Gates-Universal Gates.	3 Hours
3	**Module 3 : Combinational Logic**	**9 Hours**
3.1	Combinational Circuits – Analysis and Design Procedures - Binary Adder-Subtractor - Carry look ahead adder, BCD adder	3 Hours
3.2	Code converter - Magnitude Comparator - Decoders – Encoders.	3 Hours
3.3	Parity Generator – Multiplexers – DE multiplexers – Implementation of Boolean functions using MUX.	3 Hours
4	**Module 4 : Sequential Logic Circuits**	**9 Hours**
4.1	Sequential Circuits - Storage Elements: Latches - Flip-Flops – RS - JK – D – T - Triggering of flip-flops - race condition- Master-Slave.	3 Hours
4.2	Analysis of Clocked Sequential Circuits.	3 Hours
4.3	State Reduction and Assignment - Design Procedure- using JK, D& T.	3 Hours
5	**Module 5 : Counters and Shift Registers**	**9 Hours**
5.1	Registers - Shift Registers – SISO, PIPO, SIPO, PISO- Universal shift registers.	2 Hours

5.2	Design of Counters- Synchronous & Asynchronous Counters - up-down counter.	3 Hours
5.3	Counters- Decade counter - BCD counter - Johnson counter – Ring counter.	2 Hours
5.4	Memory & Programmable logic-RAM, ROM, PLA,PAL.	2 Hours

ASSIGNMENT QUESTIONS

Assignment -I

1. What are the character coding schemes .Write a note on ASCII and EBCDIC.

Assignment -II

1. Explain about the Read only Memory (ROM) and different types of ROM.

TUTORIAL QUESTIONS

1. Represent the decimal numbers (456)10 and (379)10 in hexadecimal and perform addition of these hexadecimal numbers.
2. Subtract (1101)2 from (11010)2 using: i) 2's complement and ii) 1'scomplement arithmetic.
3. Perform the following operations using 2's complement arithmetic: (i) (88)10 (ii) 10 + (−37)10 (−20)10 + (−12)10.
4. Perform the following base conversions: (i) (101011.11)2 to octal (ii) (3F9B)16 to binary (iii) (121)10 to binary (iv) (3077)8 to binary.
5. Convert 203.5210 to binary and hexadecimal.
6. Convert the hexadecimal number F3A7C2 to binary and octal.
7. Convert the following decimal numbers to the bases indicated.
    a. 7562 to octal
    b. 1938 to hexadecimal
    c. 175 to binary
8. Convert the gray code number 11011 to binary.
9. Represent the decimal number 3452 in i)BCD ii)Excess-3.
10. Perform (-50)-(-10) in binary using the signed-2's complement.
11. Determine the validity of DeMorgan's theorem for three variables: (ABC)' • A' + B' + C'.

12. Simplify the following expressions using Boolean algebra.
    a. A + AB
    b. AB + AB'
    c. A' BC + AC
    d. A'B +ABC' + ABC
13. Simplify the following expressions using Boolean algebra.
    a. AB + A(CD + CD')
    b. (BC' + A'D) (AB' + CD')
14. Using DeMorgan's theorem,
    show thata. (A + B)'(A' +
    B')' = 0
    b. A + A'B + A'B' =1
15. Given the Boolean function F =x'y + xyz':Derive an algebraic expression for thecomplement F'.
    a.Show that F • F' =0.
    b.Show that F + F' = I.
16. Convert the given expression in canonical SOP form Y = AC + AB + BC.
17. Convert the given expression in canonical POS form Y = (A + B)(B + C)(A + C).
18. Find the minterms of the logical expression Y = A'B'C' + A'B'C + A'BC + ABC'
19. Write the maxterms corresponding to the logical expression Y = (A + B + C')
20. Reduce AB + (AC)' + AB'C (AB + C).
21. Implement the Boolean function with three half adder circuit F = A'BC + AB'C
22. Design 8:1 MUX by using two 4:1 MUX
23. Implement the following function using 8 ∗ 1 and 16 ∗ 1 multiplexer. F (a, b, c, d) = ∈m (0, 1, 2, 3, 4, 5, 8, 9, 10, 11, 15)
24. Design JK counter that goes through states 0, 2, 3, 5, 6 only.
25. Design a J-K flip flop counter that goes through the states 1, 3, 5, 7, 1, 3.... . Is the counter self-correcting?
26. Design an up- down 3-bit synchronous counter using JK flip flop to count the following sequence 0, 2, 3, 6, 4, 0.
27. Design a Up- down synchronous counter using D flip flop to count the following sequence: 0, 5, 7, 6, 3, 2, 4, 0.
28. Design J-K counter that goes through states: 0, 3, 5, 0.

29. Design and draw the circuit of shift register to generate the following wave train .....1101011.....…
30. Design MOD-5 Up- down counter using RD flip flops.

PROBLEM SOLVING USING PYTHON

COURSE INFORMATION SHEET

PROGRAMME: INFORMATION TECHNOLOGY	**DEGREE:** BTECH
COURSE: PROBLEM SOLVING USING PYTHON	**SEMESTER:** III **CREDITS**: 4
COURSE CODE: 101004/IT300D REGULATION: 2021	**COURSE TYPE**: CORE
COURSE AREA/DOMAIN: PROGRAMMING	**CONTACT HOURS**: 3+1(Tutorial) hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY): 101004/IT322T	**LAB COURSE NAME**: PROGRAMMING AND SYSTEM UTILITES LAB

SYLLABUS:

UNIT	DETAILS	HOURS
I	**Introduction To Python:** Understanding Python-identifiers, variables, keywords, expressions and statements, evaluation of expressions, Operators and operands, operator precedence, indentation. **Python Program Flow Control:** Decision making- if, if..else, elif. Loops - for, while, for...else, while...else, Control statements using pass, continue, break.	9
II	**Strings and lists** – string traversal, string slices and comparison with examples, The string module, character classification. List- List values, accessing elements, list membership, Lists and for loops, List operations, List slices, List deletion, Matrices. Tuples - mutability and tuples, tuple assignment, Tuples as return values, Tuple operations. Dictionaries – operations and methods.	9
III	**Python Functions, Modules and Packages:** Function definition, calling functions, parameters and arguments, the return statement, type conversion and coercion, composition of functions, Lambda function, mathematical functions, user-defined functions, Recursion, Modules- Built-in modules, creating modules, import statement. Packages in Python - importing modules from a package	9
IV	**Python Files and exceptions:** Python file handling, open, write, read text files, writing variables, Directories in Python, Pickling, Exception Handling.	9
V	**Introduction to data structures in Pandas** Series and Data Frame, Operations on a Series – head, tail, vector operations, Data Frame operations: create, display, iteration, select column, add column, delete column, Binary operations in a Data Frame – add, sub, mul, div, radd, rsub, Matching and broadcasting operations, Missing data and filling values. Comparisons, Boolean reductions, comparing Series, and combining Data Frames.	9
	TOTAL HOURS	45

TEXT/REFERENCE BOOKS:

T	Allen Downey, Jeffrey Elkner, Chris Meyers, *How to think like a Computer ScientistLearning with Python*, Green Tea Press, First edition, 2002
T	Mark Lutz, *Learning Python: Powerful Object-Oriented Programming* , O"Reilly Media Inc.,5th,2013
R	Kenneth A. Lambert, B. L. Juneja, "Fundamentals of Python", Cengage Lerning India Pvt. Ltd., 2015.
R	S.A.Kulkarni, "Problem Solving and PYTHON Programming", 2nd edition, Yes
R	Dee Publishing Pvt Ltd, 2018
R	Mark Summerfield,"Programming in Python 3: A Complete Introduction to the Python Language", Pearson Education, 2nd,2018
R	Yashavant Kanetkar ,Aditya Kanetkar ,"Let Us Python ",BPB Publications,
R	1st Edition, 2019 Allen Downey, "Learning with Python", Dreamtec Press,
	1st Edition, 2015 https://docs.python.org/3/reference/

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
	NIL		

COURSE OBJECTIVES:

1	*To develop problem solving skills*
2	*To learn programming and to solve problems using Python*

COURSE OUTCOMES:

CO No.	**Course Outcomes**	**Bloom's Category**
CO 1	Write programs using Python and learn its execution environment	Understand
CO 2	Apply programs to implement various computational tasks which requires loops and conditional statements	Apply
CO 3	Write programs using functions and packages	Understand
CO 4	Apply programs to implement the concept of file handling using python	Apply
CO 5	Design object oriented programs to implement daily life problems and their	Apply

	solutions	

CO-PO MAPPING

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	1	1	1	-	-	-	-	-	-	-	-	-
CO 2	2	2	1	1	-	-	-	-	-	-	-	-
CO 3	3	2	2	2	2	-	-	-	-	-	-	1
CO 4	1	1	1	2	1	-	-	-	-	-	-	1
CO 5	3	1	3	2	1	1	-	-	-	-	-	1

CO-PSO MAPPING

COs	PSO 1	PSO 2	PSO 3
CO 1	2	-	-
CO 2	2	-	-
CO 3	2	-	-
CO 4	2	-	-
CO 5	2	-	-

JUSTIFICATION

Mapping	Level	Justification
CO1-PO1	1	Writing programs in python will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO1-PO2	1	Writing programs in python will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.

CO1-PO3	1	Writing programs in python will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO2-PO1	2	Writing programs in python programs to implement various computational tasks which requires loops and conditional statements will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO2-PO2	2	Writing programs in python programs to implement various computational tasks which requires loops and conditional statements will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
CO2-PO3	1	Writing programs in python programs to implement various computational tasks which requires loops and conditional statements will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems.
CO2-PO4	1	Writing programs in python programs to implement various computational tasks which requires loops and conditional statements will help to conduct investigations of complex problems
CO3-PO1	3	Writing programs in pythons using functions and packages will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO3-PO2	2	Writing programs in pythons using functions and packages will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
C03-PO3	2	Writing programs in pythons using functions and packages will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO3-PO4	2	Writing programs in pythons using functions and packages will help to conduct investigations of complex problems
CO3-PO5	2	Writing programs in pythons using functions and packages will help to Create, select, and apply appropriate techniques using IT tools
CO3-P12	1	Writing programs in pythons using functions and packages will help to have the preparation and ability to engage in independent and life-long learning in the broadest context of

		technological change.
CO4-PO1	1	Writing programs in python using the concept of files will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO4-PO2	1	Writing programs in python using the concept of files will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
C04-PO3	1	Writing programs in python using the concept of files will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO4-PO4	2	Writing programs in python using the concept of files will help to conduct investigations of complex problems
CO4-PO5	1	Writing programs in python using the concept of files will help to create, select, and apply appropriate techniques using IT tools
CO4-P12	1	Writing programs in python using the concept of files will help to have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
CO5-PO1	3	Writing python programs in pandas to implement daily life problems and their solutions will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO5-PO2	1	Writing python programs to implement daily life problems and their solutions will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
C05-PO3	3	Writing python programs to implement daily life problems and their solutions will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO5-PO4	2	Writing object oriented python programs to implement daily life problems and their solutions will help to conduct investigations of complex problems
CO5-PO5	1	Writing object oriented python programs to implement daily life problems and their solutions will help to create, select, and apply appropriate techniques using IT tools
CO5-P06	1	Writing python programs to implement daily life problems and their solutions will be relevant to the professional

		engineering practice
CO5-P12	1	Writing python programs to implement daily life problems and their solutions will have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Mapping	Level	Justification
CO1-PSO1	2	Writing programs in python will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO2-PSO1	2	Writing programs in python programs to implement various computational tasks which requires loops and conditional statements will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO3-PSO1	2	Writing programs in pythons using functions and packages will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO4-PSO1	2	Writing programs in python using the concept of files will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO5-PSO1	2	Writing python programs in pandas to implement daily life problems and their solutions acquire skills to design, analyse and develop algorithms and implement those using Python.

GAPS IN THE SYLLABUS - TO MEET INDUSTRY/PROFESSIONAL REQUIREMENTS:

SNO	DESCRIPTION	PROPOSED ACTIONS
1	Effective Python Programming	Special Lecture

PROPOSED ACTIONS: TOPICS BEYOND SYLLABUS/ASSIGNMENT/INDUSTRY VISIT/GUEST LECTURER/NPTEL ETC

1	Numpy module to get efficient and fast results with arrays and matrices https://www.python-course.eu/numpy.php

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

WEB SOURCE REFERENCES:

1	https://archive.org/details/MIT6.00SCS11
2	https : //www. coursera.org/course/pythonlearn

3	https://www.python.org/

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

✓ CHALK & TALK	✓ STUD. ASSIGNMENT	✓ WEB RESOURCES	
✓ LCD/SMART BOARDS	□ STUD. SEMINARS	✓ ADD-ON COURSES	

ASSESSMENT METHODOLOGIES-DIRECT

✓ ASSIGNMENTS	✓ STUD. SEMINARS	✓ TESTS/MODEL EXAMS	✓ UNIV. EXAMINATION
✓ STUD. LAB PRACTICES	✓ STUD. VIVA	□ MINI/MAJOR PROJECTS	□ CERTIFICATIONS
□ ADD-ON COURSES	□ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

✓ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	✓ STUDENT FEEDBACK ON FACULTY
□ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS	□ OTHERS

Prepared by
Ms. Jeshmol P
Faculty

Approved by
Dr. Neeba E A
HOD

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

101004/IT300D PROBLEM SOLVING USING PYTHON

S3 IT Course Plan - 2022-23

Day 1	Understanding Python-identifiers, variables	1
Day 2	keywords, expressions and statements	1
Day 3	Evaluation of expressions, Operators and operands, operator precedence, indentation	1
Day 4	Python Program Flow Control: Decision making	1
Day 5	if, if..else, elif.	1
Day 6	Loops - for, while,	1
Day 7	for...else, while...else	1
Day 8	Control statements using pass	1
Day 9	continue, break.	1
Day 10	String traversal, string slices and comparison with	2

	examples	
Day 11	The string module, character classification	2
Day 12	List- List values, accessing elements, list membership, Lists and for loops	2
Day 13	List operation, List slices, List deletion	2
Day 14	Matrices	2
Day 15	Tuples- mutability and tuples, tuple assignment, tuples as return values	2
Day 16	Tuple operations.	2
Day 17	Dictionaries - operations and methods	2
Day 18	Dictionaries - operations and methods	2
Day 19	Function definition, calling functions, parameters and arguments, the return statement.	3
Day 20	Type conversion and coercion, composition of functions	3
Day 21	Lambda function, mathematical functions	3

Day 22	User-defined functions	3
Day 23	Recursion	3
Day 24	Modules - Built-in modules	3
Day 25	Creating modules, import statement	3
Day 26	Packages in Python - importing modules from a package	3
Day 27	Packages in Python - importing modules from a package	3
Day 28	Python file handling, open, write, read text files	4
Day 29	Python file handling, open, write, read text files	4
Day 30	Python file handling, open, write, read text files	4
Day 31	Python file handling, open, write, read text files	4
Day 32	Writing variables	4
Day 33	Directories in Python	4

Day 34	Pickling	4
Day 35	Exception Handling	4
Day 36	Exception Handling	4
Day 37	Introduction to data structures in Pandas :Series and Data Frame	5
Day 38	Operations on a Series: head, tail, vector operation	5
Day 39	Data Frame operations: create, display, iteration, select column, add column, delete column	5
Day 40	Binary operations in a Data Frame - add, sub, mul, div, radd, rsub	5
Day 41	Matching and broadcasting operations	5
Day 42	Comparisons	5
Day 43	Boolean reductions	5
Day 44	Comparing Series	5
Day 45	combining Data Frames.	5

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

101004/IT300D PROBLEM SOLVING USING PYTHON

ASSIGNMENT – I

1. Give the features of python.

2. List the standard data types in python.

3.What is tuple? What is the difference between list and tuple? Give an example for their usage.

4. Give the features of python dictionaries.

5. What is a multiline statement?

6What are logical operators and Bitwise operators?

8. Give the characteristics of a membership operator?

9. Write short notes on types of operators in python with appropriate example.

10. Explain the features of a dictionary.

11. Write a short note on assert function.

12. What is the difference between lists, tuples and dictionaries? Give an example for their usage.

13. What are the basic list operations that can be performed in Python? Explain each operation with its syntax and example.

14. What is a Dictionary? Explain Python dictionaries in detail discussing its operations and methods.

15. a) Write a Python program to add “ing“ at the end of a string. If the string already ends with “ing“ then add “ly“.

b) Write a program to print an index at which a particular value exists in a list of numbers. If the value exists at multiple locations in the list, then print all the indices.

ASSIGNMENT – II

1. Discuss Modules- Built-in modules, creating modules, import statements in python.

2. Explain in detail about Packages in Python - importing modules from a package.

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

101004/IT300D PROBLEM SOLVING USING PYTHON

TUTORIAL

Pattern #1: Simple Number Triangle Pattern

Pattern:

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

Pattern #2: Inverted Pyramid of Numbers

Pattern:

1 1 1 1 1

2 2 2 2

3 3 3

4 4

5

Pattern #3: Half Pyramid Pattern of Numbers

Pattern:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Pattern #4: Inverted Pyramid of Descending Numbers

Pattern:

5 5 5 5 5

4 4 4 4

3 3 3

2 2

1

Pattern #5: Inverted Pyramid of the Same Digit

Pattern:

5 5 5 5 5

5 5 5 5

5 5 5

5 5

5

Pattern #6: Reverse Pyramid of Numbers

Pattern:

1

2 1

3 2 1

4 3 2 1

5 4 3 2 1

Pattern #7: Inverted Half Pyramid Number Pattern

Pattern:

0 1 2 3 4 5

0 1 2 3 4

0 1 2 3

0 1 2

0 1

Pattern #8: Pyramid of Natural Numbers Less Than 10

Pattern:

1

2 3 4

5 6 7 8 9

Pattern #9: Reverse Pattern of Digits from 10

Pattern:

1

3 2

6 5 4

10 9 8 7

Pattern #10: Unique Pyramid Pattern of Digits

Pattern:

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1

Pattern #11: Connected Inverted Pyramid Pattern of Numbers

Pattern:

5 4 3 2 1 1 2 3 4 5

5 4 3 2 2 3 4 5

5 4 3 3 4 5

5 4 4 5

5 5

Pattern #12: Even Number Pyramid Pattern

Pattern:

10
10 8
10 8 6
10 8 6 4
10 8 6 4 2

Pattern #13: Pyramid of Horizontal Tables

Pattern:

```
0
0 1
0 2 4
0 3 6 9
0 4 8 12 16
0 5 10 15 20 25
0 6 12 18 24 30 36
```

Pattern #14: Pyramid Pattern of Alternate Numbers

Pattern:

```
1
3 3
5 5 5
7 7 7 7
9 9 9 9 9
```

Pattern #15: Mirrored Pyramid (Right-angled Triangle) Pattern of Numbers

Pattern:

```
        1
      1 2
    1 2 3
  1 2 3 4
1 2 3 4 5
```

Pattern #16: Equilateral Triangle with Stars (Asterisk Symbol)

Pattern:

```
       *
      * *
     * * *
    * * * *
   * * * * *
  * * * * * *
 * * * * * * *
```

Pattern #17: Downward Triangle Pattern of Stars

Pattern:

```
 * * * * * *
  * * * * *
```

```
 * * * *
  * * *
   * *
    *
```

Pattern #18: Pyramid Pattern of Stars

Pattern:

```
*

* *

* * *

* * * *

* * * * *
```

DESIGN AND ENGINEERING

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

COURSE INFORMATION SHEET
101908/CO900E – DESIGN AND ENGINEERING

PROGRAMME: Information Technology / Artificial Intelligence and Data Science	DEGREE: B.Tech.
COURSE: **DESIGN AND ENGINEERING**	SEMESTER: S3 CREDITS: 2
COURSE CODE: 101908/CO900E REGULATION: 2020	COURSE TYPE: Core
COURSE AREA/DOMAIN: Engineering	CONTACT HOURS: 2 (Lecture) + 0 (Tutorial) + 0 (Practical) Hours / Week
CORRESPONDING LAB COURSE CODE (IF ANY): NIL	LAB COURSE NAME: NA

SYLLABUS:

Module	DETAILS	HOURS
I	**Design Process** Introduction to Design and Engineering Design- Defining a Design Process- Detailing Customer Requirements, Setting Design Objectives, Identifying Constraints, Establishing Functions, Generating Design Alternatives and Choosing a Design.	05
II	**Design Thinking Approach** Introduction to Design Thinking-Iterative Design Thinking Process Stages- Empathize, Define, Ideate, Prototype and Test-Design Thinking as Divergent-Convergent Questioning- Design Thinking in a Team Environment.	05
III	**Design Communication(Languages of Engineering Design)** Communicating Designs Graphicall-Communicating Designs Orally and in Writing-Mathematical Modeling In Design- Prototyping and Proofing the Design.	05
IV	**: Design Engineering Concepts** Project-based Learning and Problem-based Learning in Design.- Modular Design and Life Cycle Design Approaches-Application of Bio-mimicry, Aesthetics and Ergonomics in Design-Value Engineering,Concurrent Engineering, and Reverse Engineering in Design.	05

V	**Expediency, Economics and Environment in Design Engineering** Design for Production-Use, and Sustainability-Engineering Economics in Design.-Design Rights.-Ethics in Design	05
	TOTAL HOURS	25

TEXT/REFERENCE BOOKS:

S.	T/	AUTHORS/BOOK TITLE/PUBLICATION
1.	T	Yousef Haik, Sangarappillai Sivaloganathan, Tamer M. Shahin, Engineering Design Process, Cengage Learning, 2003, Third Edition. ISBN-10: 9781305253285
2.	T	Voland. G., Engineering by Design, Pearson India 2014, Second Edition, ISBN: 9332535051
3.	R	Philip Koshy, RobertBalmer, William Keat, George Wise, Exploring Engineering, Fourth Edition: An Introduction to Engineering and Design, 4th Edition, Academic Press, 2015, ISBN: 9780128012420.
4.	R	Clive L. Dym, Engineering Design: A Project based Introduction, John Wiley & Sons, New York 2009, Fourth Edition, ISBN: 978-1-118-32458-5
5.	R	Nigel Cross, Design Thinking: Understanding How Designers Think and Work, Berg Publishers 2011, First Edition, ISBN: 978-1847886361.
6.	R	Pahl. G., Beitz W., Feldhusen J., rote K. H., Engineering Design: A Systematic Approach, Springer 2007, Third Edition, ISBN: 978-1-84628-319-2.

COURSE PRE-REQUISITES:

Nil. The course will be generic to all engineering disciplines and will not require specialized preparation or prerequisites in any of the individual engineering disciplines.

COURSE OBJECTIVES:

1	To introduce the undergraduate engineering students the fundamental principles of design engineering.
2	To make them understand the steps involved in the design process.
3	To familiarize them with the basic tools used and approaches in design.

COURSE OUTCOMES:

Ref. No.	DESCRIPTION	Blooms Taxonomy Level
101908/ CO900E.1	Explain the different concepts and principles involved in design engineering.	Remember, Understand (Level 1 & 2)
101908/ CO900E.2	Apply design thinking while learning and practicing engineering.	Apply (Level 3)
101908/ CO900E.3	Develop innovative, reliable, sustainable and economically viable designs incorporating knowledge in engineering.	Apply (Level 3)

CO-PO AND CO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1
101908/CO900E.1	2	1					1			1			2
.2		2				1		1				2	2
.3			2			1	1		2	2		1	

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM /HIGH	JUSTIFICATION
101908/CO900E.1-PO1	M	Students will be able to apply the concepts and principles of design engineering.
101908/CO900E.1-PO2	L	Students will be able to analyze and design complex engineering problems.
101908/CO900E.1-PO7	L	Students will be able to understand the impact of engineering designs in societal and environmental contexts.
101908/CO900E.1-PO10	L	Students will be able to communicate the designs effectively to engineering community and society at large.
101908/CO900E.1-PSO1	M	Students will be able to identify, analyze and design complex engineering problems in the area of computer science.
101908/CO900E.1-PSO3	M	Students will be able to design and develop innovative products to meet the societal needs and thereby emerge as eminent researcher and entrepreneur.
101908/CO900E.2-PO2	M	Students will be able to analyze and design complex engineering problems by applying design thinking approach.
101908/CO90	L	Students will be able to apply design thinking approach considering the societal, health, safety, legal, and cultural

0E.2-PO6		issues.
101908/CO90 0E.2-PO8	L	Students will be able to apply design thinking approach while adhering to ethics and professional responsibility.
101908/CO90 0E.2-PO12	M	Students will be able to recognize the need for & engage in independent and life-long learning in the context of technological change.
101908/CO90 0E.2-PSO1	M	Students will be able to identify, analyze and design complex engineering problems in the area of computer science by applying design thinking approach.
101908/CO90 0E.2-PSO3	M	Students will be able to design and develop innovative products in the computer science area to meet the societal needs and thereby emerge as eminent researcher and entrepreneur.
101908/CO90 0E.3-PO3	M	Students will be able to design and develop solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
101908/CO90 0E.3-PO6	L	Students will be able to develop designs considering the societal, health, safety, legal, and cultural issues.
101908/CO90 0E.3-PO7	L	Students will be able to develop designs, understanding the impact of engineering designs in societal and environmental contexts.
101908/CO90 0E.3-PO9	M	Students will function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings while developing innovative designs.
101908/CO90 0E.3-PO10	M	Students will be able to communicate the designs they develop effectively to engineering community and society at large.
101908/CO90 0E.3-PO12	L	Students will be able to recognize the need for & engage in independent and life-long learning in the context of technological change while involving design and development.
101908/CO90 0E.3-PSO3	M	Students will be able to design and develop innovative products in the computer science area to meet the societal needs and thereby emerge as eminent researcher and entrepreneur.

INDUSTRY RELEVANCE:

Innovation and entrepreneurship has become a focus area in all universities and industry for more than a decade. We have seen many start-up companies and student entrepreneurs coming up with innovative products, disruptive technologies & businesses. Design and Engineering is a new course introduced by KTU in the B.Tech. programme and we are

following the same for Autonomous batch with the intention of fostering innovation and entrepreneurship among engineering students. As students do this course, they may come up with innovative ideas & concepts, design and develop those concepts as marketable products and may become entrepreneurs by the time they complete B.Tech.

Also, as engineers/designers in software/IT industry where majority of our students are going work, they would help the businesses to grow by designing and developing innovative products and services.

GAPS IN THE SYLLABUS - TO MEET INDUSTRY/PROFESSION REQUIREMENTS: None identified at this time.

S. NO	DESCRIPTION	PROPOSED ACTIONS	PO MAPPING

PROPOSED ACTIONS: NA

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

S. NO	TOPIC	PO MAPPING
1		
2		

WEB SOURCE REFERENCES:

1	https://www.interaction-design.org/
2	

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

☐ CHALK & TALK	☐ **STUD. ASSIGNMENT √**	☐ **WEB RESOURCES √**
☐ LCD/SMART BOARDS	☐ STUD. SEMINARS	☐ DISCUSSIONS/ DEBATES
☐ **ONLINE CLASSES √**	☐ **ONLINE QUIZ & POLLS √**	

ASSESSMENT METHODOLOGIES-DIRECT:

☐ **ASSIGNMENTS √**	☐ STUD. SEMINARS	☐ **TESTS/MODEL EXAMS √**	☐ **UNIV. EXAMINATION √**
☐ STUD. LAB PRACTICES	☐ STUD. VIVA	☐ MINI/MAJOR PROJECTS	☐ CERTIFICATIONS

☐ ADD-ON COURSES	☐ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT:

☐ **ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE) √**		☐ **STUDENT FEEDBACK ON FACULTY (ONCE) √**	
☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS		☐ OTHERS	
Module	**Day**	**Topic**	**Remarks**
1	1	Introduction to Design and Engineering Design	
	2	Engineering Design, Design Process	
	3	Defining a Design Process-:Detailing Customer Requirements	
	4	Design Process - Identifying Constraints, Establishing Functions	
	5	Design Process - Establishing Specifications	
	6	Design Process - Generating Design Alternatives & Choosing a Design	
		ASSIGNMENT I	
2	7	Introduction to Design Thinking	
	8	Iterative Design Thinking Process Stages: Empathize, Define, Ideate, Prototype and Test	
	9	Design Thinking Process Stages - Empathize (Contd..), Analyze, Define, Ideate, Prototype and Test	
	10	Design Thinking as Divergent-Convergent Questioning.	
	11	Design Thinking in a Team Environment	
	12	**Assignment Presentation**	
3	13	Communicating Designs Graphically	
	14	Communicating Designs Orally and in Writing	
	15	Mathematical Modelling in Design	
	16	Prototyping and Proofing the Design	
	17	Case Studies: Communicating Designs Graphically	
4	18	Project based Learning and Problem based Learning in Design	

	19	Modular Design and Life Cycle Design Approaches	
	20	Application of Bio-mimicry, Aesthetics and Ergonomics in Design	
	21	Value Engineering, Concurrent Engineering, and Reverse Engineering in Design	
	22	Case Studies: Bio-mimicry based Designs	
		ASSIGNMENT II	
5	23	Design for Production, Use and Sustainability	
	24	Engineering Economics in Design	
	25	Design Rights	
	26	Ethics in Design	
	27	Case Studies: Design for Production, Use and Sustainability	

Assignment 1

Students are divided into 12 groups of 6 each in their roll no. order and let each group go through different stages of design process for designing the simple products/solutions for one of the problems listed below.

- Develop the following artifacts from different stages of design process:
  i. Need/Problem Statement
  ii. Prioritized list of requirements
  iii. Objective Tree
  iv. List of Design Constraints
  v. Functional Structure
  vi. Morphological Chart
  vii. Decision Matrix & Selection of Best Solution(s)
  viii. Conceptual Sketches for 2 Best Alternate Designs Chosen.

List of Creative Design Problems to solve:

1. List out the problems of present design of safety pins. Suggest alternative designs to overcome the problems.
2. Design modifications for a nail cutter to avoid nails falling on the floor while cutting
3. Non-availability of people for cutting tree branches and plucking coconuts
4. School children have to carry extra heavy bags daily to school. Design an alternative for school bags to carry the heavy book loads.
5. Design modifications for wipers in a car so that visibility through car screen is enhanced during rainy season.
6. Design a product which can be used as a roller skate as well as a bicycle.
7. Modify the design of a cloth hanger to increase its usability and marketability.
8. Suggest new designs for a pair of shoes with a material other than the ones available in the market at present, so that its durability and value can be increased.
9. Design a kitchen tool that combines the functions of a scissor, knife and vegetable cutter/scrapper.
10. Design modifications for a wheel chair so that it can be used for climbing and descending stairs.
11. Suggest alternative designs for thermo flask especially to carry insulin vial or pen while travelling.
12. Suggest alternative designs (other than rain coats) to protect two wheeler riders from rain.
13. Middle-aged people have difficulty in standing for long time (especially in church, buses and other public places). Design a handy chair that can be carried easily.

14. Design a multi-purpose walking stick for aged people.
15. List out the limitations of berths in Indian railway coaches. Suggest alternative designs to overcome these limitations.
16. Suggest ideas/solutions to detect the presence of letters/materials in your mail box at the gate.
17. Rajagiri campus may get flooded every year during rainy season. Suggest ideas/solutions to avoid flooding and/or to reduce the impact/consequences.
18. Identify different ways to reduce manpower for cleaning / gardening etc. in your college.
19. Identify ideas/solutions to prevent dust falling from blackboard duster while using it.
20. Identify alternative ideas/solutions for cleaning ceiling fans.
21. Suggest ideas/solutions for 'detecting a golf ball'.
22. Students always forget to bring their id cards to college. Identify alternative ideas/solutions to overcome this problem.
23. Taking attendance is a time consuming process now. Suggest alternative ideas/solutions for making it easy and fast.
24. Identify additional features for a carom board to increase its value.

Assignment 2

Prepare a poster for a product which will specify the different features based on DFX or make a model of a product.

SUSTAINABLE ENGINEERING

COURSE INFORMATION SHEET

PROGRAMME:	DEGREE: B. TECH
COURSE: SUSTAINABLE ENGINEERING	SEMESTER: 3 CREDITS: NIL
COURSE CODE: 101908/CO300F REGULATION: 2022	COURSE TYPE: NON- CORE
COURSE AREA/DOMAIN: HUMANITIES	CONTACT HOURS: 2 hrs.
CORRESPONDING LAB COURSE CODE (IF ANY): Nil	LAB COURSE NAME: Nil

SYLLABUS:

Module 1: Sustainability

Introduction, concept, evolution of the concept; Social, environmental and economic sustainability concepts; Sustainable development, Nexus between Technology and Sustainable development; Millennium Development Goals (MDGs) and Sustainable Development Goals (SDGs), Clean Development Mechanism (CDM).

Module 2: Environmental Pollution

Air Pollution and its effects, Water pollution and its sources, Zero waste concept and 3 R concepts in solid waste management; Greenhouse effect, Global warming, Climate change, Ozone layer depletion, Carbon credits, carbon trading and carbon foot print, legal provisions for environmental protection.

Module 3: Environmental management standards

ISO 14001:2015 frame work and benefits, Scope and goal of Life Cycle Analysis (LCA), Circular economy, Bio-mimicking, Environment Impact Assessment (EIA), Industrial ecology and industrial symbiosis.

Module 4: Resources and its utilisation

Basic concepts of Conventional and non-conventional energy, General idea about solar energy, Fuel cells, Wind energy, Small hydro plants, bio-fuels, Energy derived from oceans and Geothermal energy.

Module 5: Sustainability practices

Basic concept of sustainable habitat, Methods for increasing energy efficiency in buildings, Green Engineering, Sustainable Urbanisation, Sustainable cities, Sustainable transport.

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T1	Allen, D. T. and Shonnard, D. R., Sustainability Engineering: Concepts, Design and Case Studies, Prentice Hall.
T2	Bradley. A.S; Adebayo,A.O., Maria, P. Engineering applications in sustainable design and development, Cengage learning
R1	Environment Impact Assessment Guidelines, Notification of Government of India, 2006
R2	Ni bin Chang, Systems Analysis for Sustainable Engineering: Theory and Applications, McGraw-Hill Professional.
R3	Twidell, J. W. and Weir, A. D., Renewable Energy Resources, English Language Book Society (ELBS).
R4	Mackenthun, K.M., Basic Concepts in Environmental Management, Lewis Publication, London, 1998

COURSE PRE-REQUISITES: NIL

--	Basic Sciences- Physics, Chemistry, Biology, Geography (High School Level)

COURSE OBJECTIVES:

1	To inculcate in students an awareness of environmental issues and the global initiatives towards attaining sustainability. The student should realize the potential of technology in bringing in sustainable practices.

COURSE OUTCOMES:

Sl. No	DESCRIPTION
1	Understand the relevance and the concept of sustainability and the global initiatives in this direction
2	Explain the different types of environmental pollution problems and their sustainable solutions
3	Discuss the environmental regulations and standards.
4	Outline the concepts related to conventional and non-conventional energy
5	Demonstrate the broad perspective of sustainable practices by utilizing engineering knowledge and principles

CO-PO-PSO MAPPING

CO No.	Programme Outcomes (POs)												Programme-specific Outcomes (PSOs)		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
1						2	3					2			2
2						2	3					2			2
3						2	3					2			2
4						2	3					2			2
5						2	3					2			2
ECT205						2	3					2			**2**

JUSTIFICATION FOR THE CORRELATION LEVEL ASSIGNED IN EACH CELL OF THE TABLE

	PO6	**PO7**	**PO12**	**PSO3**
CO1	The knowledge about the concept and importance of sustainability will help the student to focus better on societal, health, safety and cultural aspects of his/her profession	The course entirely deals with environment and sustainability, and thus all the course outcomes fully contributes to this programme outcome	Sustainable engineering is one of the elements of ethical engineering practices.	Each of the COs creates an awareness in the student about carrying out their responsibilities with due the consideration towards environment protection and sustainability.
CO2	Student's understanding of causes, effects and control of pollution contributes to making him/her a responsible engineer		Provides scope for implementation of strategies to curb environmental pollution	
CO3	Student's basic knowledge of environmental standards and environmental impact assessment will guide him/her in the assessment of his/her engineering practice.	The course entirely deals with environment and sustainability, and thus all the course outcomes fully contributes to this programme outcome	Leads to more efficient energy management systems based on EIA	
CO4	The understanding of basic concepts on conventional and non-conventional energy sources will enable the student to understand importance of energy efficient systems		Leads to more efficient utilization of resource and energy consumption	
CO5	The student's understanding of sustainable development will help him be a responsible engineer working for the benefit of the society.		Helps the student to opt for sustainable energy resources where applicable in the project.	

GAPS IN THE SYLLABUS - TO MEET INDUSTRY/PROFESSION REQUIREMENTS:

Sl No	DESCRIPTION	PROPOSED ACTIONS	PO MAPPING
1	Practical Case studies	Discussions, Presentations	PO6, PO7, PO12

PROPOSED ACTIONS: TOPICS BEYOND SYLLABUS/ASSIGNMENT/INDUSTRY VISIT/GUEST LECTURER/NPTEL ETC

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

Sl No	DESCRIPTION	PO MAPPING
1	Group Discussions	PO6, PO7, PO9, PO10
2	Seminars & Projects	PO6, PO7, PO9, PO10

WEB SOURCE REFERENCES:

1	http://www.pittstate.edu/office/president/initiatives/sustainability/what-is-sustainability.dot
2	http://www.un.org/documents/ga/conf151/aconf15126-1annex1.htm
3	http://unfccc.int/kyoto_protocol/mechanisms/
4	http://www.epa.gov/
5	http://ecometrica.com/assets/whatis_acarbonfootprint_summary.pdf

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

~~☐ CHALK & TALK~~	☐ STUD. ASSIGNMENT	☐ WEB RESOURCES
☐LCD/SMART BOARDS	☐STUD. SEMINARS	~~☐ ADD-ON COURSES~~

ASSESSMENT METHODOLOGIES-DIRECT

☐ ASSIGNMENTS	☐STUD. SEMINARS	☐ TESTS/MODEL EXAMS	☐ UNIV. EXAMINATION
☐ ~~STUD. LAB PRACTICES~~	~~☐ STUD. VIVA~~	~~☐ MINI/MAJOR PROJECTS~~	~~☐ CERTIFICATIONS~~
☐~~ADD-ON COURSES~~	☐ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

☐ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	☐ STUDENT FEEDBACK ON FACULTY
~~☐ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS~~	~~☐ OTHERS~~

Prepared by **Approved by**

Dr.Biju Paul (HoD)

Course Plan

No	Topic	No. of Lectures
1	**Sustainability**	
1.1	Introduction, concept, evolution of the concept	1
1.2	Social, environmental, and economic sustainability concepts	1
1.3	Sustainable development, Nexus between Technology and Sustainable	1
1.4	Millennium Development Goals (MDGs) and Sustainable Development Goals (SDGs)	1
1.5	Clean Development Mechanism (CDM)	1
2	**Environmental Pollution**	
2.1	Air Pollution and its effects	1
2.2	Water pollution and its sources	1
2.3	Zero waste concept and 3 R concepts in solid waste management	1
2.4	Greenhouse effect, Global warming, Climate change, Ozone layer depletion	1
2.5	Carbon credits, carbon trading and carbon footprint.	1
2.6	Legal provisions for environmental protection.	1
3	**Environmental management standards**	
3.1	Environmental management standards	1
3.2	ISO 14001:2015 framework and benefits	1
3.3	Scope and Goal of Life Cycle Analysis (LCA)	1
3.4	Circular economy, Bio-mimicking	1
3.5	Environment Impact Assessment (EIA)	1
3.6	Industrial Ecology, Industrial Symbiosis	1

4	**Resources and its utilisation**	
4.1	Basic concepts of Conventional and non-conventional energy	1
4.2	General idea about solar energy, Fuel cells	1
4.3	Wind energy, Small hydro plants, biofuels	1
4.4	Energy derived from oceans and Geothermal energy	1
5	**Sustainability Practices**	
5.1	Basic concept of sustainable habitat	1
5.2	Methods for increasing energy efficiency of buildings	1
5.3	Green Engineering	1
5.4	Sustainable Urbanisation, Sustainable cities, Sustainable transport	1

DATA STRUCTURES LAB

COURSE INFORMATION SHEET

PROGRAMME: **Computer Science/Information Technology/Artificial Intelligence**	DEGREE: **BTECH**
COURSE: **DATA STRUCTURES LAB**	SEMESTER: **III** CREDITS: **1**
COURSE CODE**:** **101903/CO322S** REGULATION: **2021**	COURSE TYPE: **CORE**
COURSE AREA/DOMAIN: **Programming, Data Structures and Algorithms**	CONTACT HOURS: **3 hours per week**

SYLLABUS:

DETAILS
1.Implementation of Polynomials and Sparse matrices using arrays** 2. Implementation of Stack , Queues, Priority Queues, DEQUEUE and Circular Queues using arrays** 3. Application problems using stacks: Conversion of expression from one notation to another notation . ** 4. Implementation of various linked list operations. ** 5. Implementation of stack, queue and their applications using linked list 6. Implementation of trees using linked list 7. Representation of polynomials using linked list, addition and multiplication of polynomials. ** 8. Implementation of binary trees using linked lists and arrays- creations, insertion, deletion and traversal. ** 9. Implementation of binary search trees – creation, insertion, deletion, search 10. Any application programs using trees 11. Implementation of sorting algorithms – bubble, insertion, selection, quick, merge sort and heap sort.** 12. Implementation of searching algorithms – linear search, binary search.** 13. Representation of graphs and computing various parameters (in degree, out degree etc.) - adjacency list, adjacency matrix. 14. Implementation of BFS and DFS for each graph representations. ** 15. Implementation of hash table using your own mapping functions and observe collisions and overflow resolving schemes. ** ** mandatory.

Lab Cycle (2023-2024)

Day 1

Revision of programs done during the C Programming Lab focussing on one dimensional and two dimensional arrays, structures, pointers and pointer to structure

1. Write a menu driven C program to implement polynomials using arrays and perform polynomial addition.

Day 2

2. Write a menu driven C program to implement sparse matrix using arrays and perform the following operations
    a. Sparse Matrix Addition
    b. Sparse Matrix Transpose

Day 3

3. Write a menu driven C program to implement stack using arrays and perform the operations on the stack (i) Push (ii) Pop (iii) Is empty (iv) Isfull (v) Display
4. Write a menu driven C program to covert infix to postfix expression and evaluate it using stack.

Day 4

5. Write a menu driven C program to implement circular queue using array and perform the following operations
   (i) Enqueue (ii) Dequeue (iii) Is empty () (iv) Is full () (v) Display

6. Write a menu driven C program to implement Double Ended Queue using array and perform the following operations
    a. Insert from the front
    b. Insert from rear
    c. Delete from front
    d. Delete from rear
    e. Display
7. Write a menu driven C program to implement Queue using arrays and perform the following operations (i) Insert (ii) Delete (iii)Is empty() (iv) Is full() (v) Display

(HA1)

Day 5

8. Write a menu driven C program to implement the following operations on a singly linked list
   (i) Insertion
    a. Insert at the beginning
    b. Insert at the end
    c. Insert after a specified node
   (ii) Deletion

a. Delete from the beginning
b. Delete from the end
c. Delete a specified node

(iii) Display

9. Write a menu driven C program to implement stack and queue using singly linked list (HA2)

Day 6

10. Write a menu driven C program to represent polynomials using linked list and perform (i) polynomial addition and (ii) polynomial multiplication.

11. Write a menu driven C program to implement a doubly linked list and perform the following operations on it
    a. Insertion (at the beginning, at the end, after a specified node)
    b. Deletion (at the beginning, at the end , a specified node)
    c. Display (Forward and Backward) (HA3)

Day 7

12. Write a menu driven C program to implement a binary search tree (BST) using singly linked list and perform the following operations
    a. Insertion
    b. Deletion
    c. Traversals
    d. Search for a specified node

Day 8

13. Write a menu driven C program to implement the following sorting techniques
    a. Bubble Sort
    b. Insertion Sort
    c. Selection Sort

14. Write a C program to implement Quick Sort

Day 9

15. Write a menu driven C program to implement Merge sort
16. Write a menu driven C program to implement searching algorithms –
    a. Linear search
    b. Binary search. (HA4)

Day 10

17. Write a C program to implement heap sort
18. Write a menu driven C program to implement hash table and the following collision resolution techniques-(i) Linear Probing (ii) Quadratic Probing (HA5)

Day 11

19. Write a menu driven C program to perform the following operations on a directed graph
    (i) DFS
    (ii) BFS
    (iii) Display (using Adjacency List and Adjacency Matrix).

**HA –Home Assignment (The programs have to be done from home and output to be verified in the lab on the date specified by the Lab In Charge. They are to be included in the record also.)*

TEXT/REFERENCE BOOKS:

Text Books

1. Horowitz and Sahni, Fundamentals of data structures - Computer Science Press
2. Gottfried B.S., Programming with C, Schaum Series, Tata McGraw Hill.

References:

1. Gary J. Bronson, ANSI C Programming, CENGAGE Learning India.
2. Stewart Venit and Elizabeth Drake, Prelude to Programming – Concepts & Design, Pearson.
3. Dromy R.G., How to Solve it by Computer, Pearson.
4. Kernighan and Ritche D.M., The C. Programming Language, PHI

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
101903/CO200G	Programming In C	The basics of C programming	S2

COURSE OBJECTIVES:

1	To give hands-on experience for Learners on creating and using different Data Structures

COURSE OUTCOMES:

SLNO	DESCRIPTION	Blooms' Taxonomy Level
101903/CO322S.1	Write a time/space efficient program using arrays/linked lists/trees/graphs to provide necessary functionalities meeting a given set of user requirements	**Cognitive Knowledge Level: Analyse**

101903/CO322S.2	Write a time/space efficient program to sort a list of records based on a given key in the record	**Cognitive Knowledge Level: Apply**
101903/CO322S.3	Examine a given Data Structure to determine its space complexity and time complexities of operations on it	**Cognitive Knowledge Level: Apply**
101903/CO322S.4	Design and implement an efficient data structure to represent given data	**Cognitive Knowledge Level: Apply**
101903/CO322S.5	Write a time/space efficient program to convert an arithmetic expression from one notation to another	**Cognitive Knowledge Level: Apply**

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
101903/CO322S.1	3	2	3	3	-	2	-	1	-	1	-	2	3	3	2
101903/CO322S.2	3	2	3	2	-	-	-	1	-	1	-	2	2	2	-
101903/CO322S.3	2	2	3	2	-	-	-	1	-	1	-	2	2	2	-
101903/CO322S.4	2	2	3	2	-	-	-	1	-	1	-	2	1	2	-
101903/CO322S.5	1	1	2	-	-	-	-	1	-	1	-	1	-	2	-
101903/CO322S overall	2	2	2	2	-	2	-	1	-	1	-	2	2	2	2

1-Low(L) 2-Medium(M) 3-High(H)

JUSTIFICATIONS FOR THE MAPPING

Mapping	LOW/MEDIUM/HIGH	Justification
101903/CO322S.1-PO1	MEDIUM	The ability of students to identify a problem and finding a solution using C programming language is a core fundamental concept needed for any Computer programmer
101903/CO322S.1-PO2	LOW	The ability of implementing solutions helps the students to analyze and propose solutions to complex engineering problems.
101903/CO322S.1-PO3	HIGH	The ability to implement a solution using C program is necessary for design/development of soutions
101903/CO322S.1-PSO1	MEDIUM	The ability of students to identify a problem and finding a solution using C programming language is a Computer Science specific skill for complex engineering problems.
101903/CO322S.1-PSO2	MEDIUM	The knowledge of C programming helps in programming and software development skills
101903/CO322S.1-PSO3	MEDIUM	The ability of students to identify a problem and finding a solution using C programming language helps in research and also in developing innovative products.
101903/CO322S.2-PO3	HIGH	The knowledge about implementing user defined data types helps in development of solutions to complex application problems.
101903/CO322S.2-PSO1	MEDIUM	The basic knowledge of implementing user defined data types is a CS specific skill for solutions to problems in vivid areas.
101903/CO322S.2-PSO2	MEDIUM	The basic knowledge of implementing user defined data types is a fundamental concept that helps students to enhance programming efficiency and software development skills
101903/CO322S.3-PO1	MEDIUM	The knowledge about implementing programs using functions and pointers is fundamental in developing efficient solutions to complex problems.
101903/CO322S.3-PO3	HIGH	The knowledge about implementing programs using functions and pointers is helpful to design/develop efficient solutions to complex problems.
101903/CO322S.3-PSO2	MEDIUM	The knowledge about implementing programs using functions and pointers helps to improve the programming efficiency.

101903/CO 322S.4-PO1	LOW	The ability of students to write programs using files and command line arguments is a fundamental concept in solving complex problems.
101903/CO 322S.4-PO3	HIGH	The ability of students to write programs using files and command line arguments helps to design/ develop solutions to complex problems.
101903/CO 322S.4-PSO1	LOW	The ability of students to write programs using files and command line arguments is a CS specific skill that helps students to write efficient C programs for solutions to various problems.
101903/CO 322S.4-PSO2	MEDIUM	The ability of students to write programs using files and command line arguments enhances the students' programming efficiency.
101903/CO 322S.5-PO1	LOW	The ability of students to implement fundamental sorting and searching techniques is a core fundamental concept which can be used for solving complex problems
101903/CO 322S.5-PO2	LOW	The ability of students to implement fundamental sorting and searching techniques helps to analyze or compare solutions and reach conclusions.
101903/CO 322S.5-PO3	MEDIUM	The ability of students to implement fundamental sorting and searching techniques helps in designing/ developing solutions to complex engineering problems.
101903/CO 322S.5-PSO2	MEDIUM	The ability of students to implement fundamental sorting and searching techniques helps in designing/ developing solutions to complex engineering problems and thereby enhancing the programming efficiency.

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

SL NO	DESCRIPTION	PROPOSED ACTIONS	RELEVANCE WITH POs	RELEVANCE WITH PSOs
1	Implementation of AVL Trees	Assign to a team of best performers as C programmers	PO3	PSO2

WEB SOURCE REFERENCES:

1	http://www.tutorialspoint.com/cprogramming/
2	https://www.coursera.org/specializations/c-programming
3	https://onlinecourses.nptel.ac.in/noc22_cs40/preview

4	http://www.w3schools.in/c-programming-language/intro/

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

✓ CHALK & TALK	✓ HOME ASSIGNMENT	✓ WEB RESOURCES
LCD/SMART BOARDS	STUD. SEMINARS	ADD-ON COURSES

ASSESSMENT METHODOLOGIES-DIRECT

ASSIGNMENTS	STUD. SEMINARS	✓ TESTS/MODEL EXAMS	✓ UNIV. EXAMINATION
✓ STUD. LAB PRACTICES	✓ STUD. VIVA	☐ MINI/MAJOR PROJECTS	☐ CERTIFICATIONS
ADD-ON COURSES	OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

✓ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	✓ STUDENT FEEDBACK ON FACULTY (ONCE)
☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS	☐ OTHERS

Prepared by
Tinku Soman Jacob

Approved by
HOD

Open Questions

1. Write a menu driven C program to implement circular queue using arrays.
2. Write a menu driven C program to implement DEQUEUE using arrays.
3. Write a program to implement sparse matrix transpose
4. Write a menu driven C program to implement a doubly linked list and perform the following operations on it:

(i) Insertion (at the beginning, at the end, after a specified node).
(ii) Deletion (at the beginning, end of a specified node).
(iii) Display (Forward and Backward).

5. Implement a queue and reverse the order of the queue using

(i) Two additional stacks
(ii) One additional queue

6. Implement a singly linked list and perform the following

(i) Remove all duplicate elements from the list
(ii) Make a copy of the given list
(iii) Remove the first and last occurrence of the given element from the list

7. Write a program to create a binary tree and determine the following

- The number of nodes in the tree
- The sum of contents of all nodes in the tree
- The depth of the tree

Advanced Questions

1. Implement two linked list L1 and L2 and find the following
   a) L1U L2
   b) L1UL2
2. Write a program to find (A+B)*(A-B), where A and B are polynomials.

a) using Linked list
b) using Arrays

3. Implement a singly linked list and perform the following

- Remove all duplicate elements from the list
- Make a copy of the given list
- Remove the first and last occurrence of the given element from the list

4. Implement two singly linked list and perform the following

- Concatenate two lists
- Form a list containing the union of elements of the two lists.

5. Implement a singly linked list and perform the following

- Delete every second element from the list
- Place the elements of a list in increasing order
- Return the sum of integers in a list

6. Write a program to create a binary tree and determine the following

- The number of nodes in the tree
- The sum of contents of all nodes in the tree
- The depth of the tree

7. Write a program to create a binary tree and determine whether the left or right child from the root is big and by how much.
8. Implement an Expression tree. Also evaluate the expression.
9. Write a program to make the copy of a binary tree.
10. Write a program to create the mirror image of the tree.

11. Write a program to create a binary tree and determine the following

- Count the number of leaf nodes
- Count the number of nodes in the left sub tree
- Count the number of nodes in the right sub tree

PROGRAMMING AND SYSTEM UTILITIES LAB

COURSE INFORMATION SHEET

PROGRAMME: INFORMATION TECHNOLOGY	**DEGREE:** BTECH
COURSE: PROGRAMMING AND SYSTEM UTILITIES LAB	**SEMESTER:** III **CREDITS:** 2
COURSE CODE: 101004/IT322T REGULATION: 2021	**COURSE TYPE:** CORE
COURSE AREA/DOMAIN: PROGRAMMING	**CONTACT HOURS:** 3 hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY):NIL	LAB COURSE NAME:NIL

SYLLABUS:

UNIT	DETAILS	HOURS
I	**Part A : Programming in Python** 1. **Basic programming experiments** to familiarization of data types and input- output statements 2. **Decision making, branching and looping statements** 3. **Function & Function calls** a) Function definitions and access b) Parameters and arguments c) Recursion **4. Strings** a) String traversal, join, slicing b) String searching, Comparison c) Other important String methods **5. Lists, Tuples and Dictionaries** a) Creation of List & List Operations b) Tuple and Tuple operations c) Creation of Dictionary and Operations d) Comparison of List and Tuple **6. Matrix representation** a) Creating matrix b) Matrix operations - addition, subtraction and multiplication **7. Files and Operations** a) Files - defining, opening/closing, read/write operations b) Exceptions in Python c) Pickling **8. Data structures in Pandas** a) Operations on Python Data Series b) Operations on Python Data Frames	

	c) Binary operations in a Data Frames d) Missing data and filling values **<u>Part B : System Utilities</u>** **Basic Windows/Linux Administration Utilities** 1. **Experiments on Windows Operating System**	

	a. Perform the following commands **DIR, TYPE, DEL, ERASE, MD, CD, COPYCON, RMDIR, REN, VER, DATE, TIME, TREE, PATH, CLS, RMDIR, BREAK, SET, EXIT, APPEND, CHKDISK, ATTRIB, SYS, EDIT, XCOPY, DISKCOPY** b. Explore and describe some system utility like **regedit**, memory partioning, control panel and window tools **2. Experiments on Linux Operating System** a) Perform general purpose utilities in Linux: **echo, uname, whoami, passwd, date, date +%T, date +%h, date +%m, date +%y, date +"%h%y", cal, cal 12 2030, echo $HOME, pwd, ls, ls –all, ls –l, cat, cat > file1, cat >> file2, ls -l >fileinfo** b) Familiarize working with files and managing file attributes **3. Network Configuration Utilities** a) **ifconfig** utility, enable/disable network interface, **traceroute**, **telnet**, **nslookup, netstat, w, scp**, etc b) Connecting to the internet **4. GIT for version control** a. Installation and configuration of Git on Ubuntu and Windows operating systems b. Perform Basic Git Commands (**gitinit, add, status, commit**, and **log**) and Git **checkout** command	

TEXT/REFERENCE BOOKS:

T / R	BOOK TITLE/AUTHORS/PUBLICATION
T	1. Allen Downey, Jeffrey Elkner, Chris Meyers, " How to think like a Computer Scientist-Learning with Python", Green Tea Press, First edition, 2002. 2. Mark Lutz,"Learning Python: Powerful Object-Oriented Programming" , O'Reilly Media Inc.,5th,2013
R	1. S.A.Kulkarni, "Problem Solving and PYTHON Programming", 2nd edition, Yes Dee Publishing Pvt Ltd, 2018 2. Kenneth A. Lambert, B. L. Juneja, "Fundamentals of Python", Cengage Learning India Pvt. Ltd., 2015. 3. Mark Summerfield," Programming in Python 3: A Complete Introduction to the Python Language", Pearson Education, 2nd,2018 5. Yashavant Kanetkar ,Aditya Kanetkar ,"Let Us Python ",BPB Publications, 1st Edition, 2019 6. Allen Downey, "Learning with Python", Dreamtec Press, 1st Edition, 2015 7. https://docs.python.org/3/reference/

	8. Version Control with Git: Powerful tools and techniques for collaborative software development 2nd Edition, Kindle Edition by Jon Loeliger, Matthew McCullough

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
	-	Python Programming Knowledge	

COURSE OUTCOMES:

CO No.	Course Outcomes	Bloom's Category
CO 1	Develop readable* Python programs by making use of basic constructs- Decision controls, Looping controls, Lists, Tuple and Strings	Create
CO 2	Design modular Python programs using normal and recursive functions	Create
CO3	Design programs using Dictionaries and Files	Create
CO 4	Experiment with the basic Windows/ Linux administration & network configuration utilities	Apply

CO 5	Experiment with version control tools using git	Apply
readable* - readability of a program means the following: 1. Logic used is easy to follow 2. Standards to be followed for indentation and formatting 3. Meaningful names are given to variables 4. Concise comments are provided wherever needed		

CO-PO MAPPING:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO 1	2	2	3	1	3	-	-	-	-	-	-	-
CO 2	2	3	3	3	3	-	-	-	-	-	-	-
CO 3	2	3	3	3	3	-	-	-	-	-	-	-
CO 4	2	2	1	2	1	-	-	-	-	-	-	-
CO 5	-	-	2	-	3	-	-	-	3	2	-	-

COs	**PSO 1**	**PSO 2**	**PSO 3**
CO 1	2	-	-
CO 2	2	-	-
CO 3	2	-	-
CO 4	1	-	-
CO 5	1	-	-

Mapping	**Level**	**Justification**
CO1-PO1	2	Writing programs in python using basic constructs will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO1-PO2	2	Writing programs in python using basic constructs will help to identify, formulate, and analyze complex engineering problems

		reaching substantiated conclusions using first principles of engineering sciences.
CO1-PO3	3	Writing programs in python using basic constructs will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO1-PO4	1	Writing programs in python using basic constructs will help to conduct investigations of complex problems
CO1-PO5	3	Writing programs in python using basic constructs will help to apply IT tools to model complex engineering activities.
CO2-PO1	2	Writing modular programs in python to implement various computational tasks will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO2-PO2	3	Writing modular programs in python to implement various computational tasks will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
CO2-PO3	3	Writing modular programs in python to implement various computational tasks will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems.
CO2-PO4	3	Writing modular programs in python to implement various computational tasks will help to conduct investigations of complex problems
CO2-PO5	3	Writing modular programs in python to implement various computational tasks will help to apply IT tools to model complex engineering activities.
CO3-PO1	2	Writing programs in pythons using dictionaries and files will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO3-PO2	3	Writing programs in pythons using dictionaries and files will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
CO3-PO3	3	Writing programs in pythons using dictionaries and files will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO3-PO4	3	Writing programs in pythons using dictionaries and files will help to conduct investigations of complex problems

CO3-PO5	3	Writing programs in pythons using dictionaries and files will help to Create, select, and apply appropriate techniques using IT tools
CO4-PO1	2	Experimenting with the basic Windows/ Linux administration & network configuration utilities will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO4-PO2	2	Experimenting with the basic Windows/ Linux administration & network configuration utilities will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
C04-PO3	1	Experimenting with the basic Windows/ Linux administration & network configuration utilities will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO4-PO4	2	Experimenting with the basic Windows/ Linux administration & network configuration utilities will help to conduct investigations of complex problems
CO4-PO5	1	Experimenting with the basic Windows/ Linux administration & network configuration utilities will help to create, select, and apply appropriate techniques using IT tools
C05-PO3	2	Experimenting with version control tools will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO5-PO5	3	Experimenting with version control tools will help to create, select, and apply appropriate techniques using IT tools.
CO5-P09	3	Experimenting with version control tools will help to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
CO5-P10	2	Experimenting with version control tools will help to Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Mapping	**Level**	**Justification**
CO1-PSO1	2	Writing programs in python using basic constructs will help to acquire skills to design, analyse and develop algorithms

		and implement those using Python.
CO2-PSO1	2	Writing modular programs in python programs to implement various computational tasks will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO3-PSO1	2	Writing programs in pythons using dictionaries and files will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO4-PSO1	1	Experimenting with operating system commands will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO5-PSO1	1	Experimenting with version control tools to implement daily life problems and their solutions acquire skills to design, analyse and develop algorithms and implement those using Python.

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

Si NO	DESCRIPTION	PROPOSED ACTIONS	RELEVANCE WITH POs	RELEVANCE WITH PSOs
1	Understanding Program Efficiency	NPTEL	3,5	1

WEB SOURCE REFERENCES:

1	https://www.programiz.com/python-programming
2	https://pythonprogramming.net/introduction-to-python-programming/
3	https://www.tutorialspoint.com/python/

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

✓ CHALK & TALK	✓ STUD. ASSIGNMENT	✓ WEB RESOURCES	
✓ LCD/SMART BOARDS	✓ STUD. SEMINARS	□ ADD-ON COURSES	

ASSESSMENT METHODOLOGIES-DIRECT

✓ ASSIGNMENTS	✓ STUD. SEMINARS	✓ TESTS/MODEL EXAMS	✓ UNIV. EXAMINATION
✓ STUD. LAB PRACTICES	✓ STUD. VIVA	□ MINI/MAJOR PROJECTS	□ CERTIFICATIONS
□ ADD-ON COURSES	□ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

✓ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	✓ STUDENT FEEDBACK ON FACULTY (TWICE)
□ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS	□ OTHERS

Prepared by

Ms. Jeshmol P J
Faculty

Approved by

Dr. Neeba E A
HOD

RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
COURSE PLAN
101004/IT322T PROGRAMMING AND SYSTEM UTILITIES LAB

DAY	EXPERIMENTS
2-11-2022	EXPERIMENT 1
9-11-2022	
16-11-2022	EXPERIMENT 2
23-11-2022	
30-11-2022	EXPERIMENT 3
7-12-2022	
14-12-2022	EXPERIMENT 4
21-12-2022	
4-01-2023	EXPERIMENT 5
11-01-2022	
18-01-2022	PRACTICE LAB
25-01-2022	MODEL EXAM

101004/IT322T PROGRAMMING AND SYSTEM UTILITIES LAB

S3 IT

Batch 2021-25

LAB CYCLE

EXPERIMENT 1

Develop a python program to

1. Print all prime numbers with in an interval
2. Search an element in a list
3. Input a list of n numbers. Calculate and display the average of numbers. Also display the square of each value in the list.
4. Add two matrices.
5. Find the number of occurrences of a given substring in a string.
6. Count the number of vowels, consonants, words and question marks in a given string.
7. To write a python program to perform Matrix Multiplication.

EXPERIMENT 2

Develop a python program to

1. Find the value of nCr using the function.
2. Implements a calculator with functions like add, subtract, multiply, divide, exponent etc.
3. Find the factorial of a given number using recursion.
4. Find nth Fibonacci number using recursion.

EXPERIMENT 3

1. Develop a python program to create a dictionary of phone numbers and names of n persons. Display the contents of the dictionary in alphabetical order of their names
2. Develop a python program to implement the following scenario. A book shop details the Title of the book and Number of copies of each title. As books are added to the shop, the number of copies in each should increase and as books are sold the number of copies in each should decrease.
3. Develop a python code to read a text file, copy the contents to another file after removing the blank lines.
4. Find the most frequent words in a text read from a file.
5. Develop a python program to implement the following scenario. Given a give “data.txt” with three columns of data separated by spaces. Read it into 3 separate simple sequences.
6. Create a class student with attributes name, rollno and a method showData() for showing the details. Create two instances of the class and call the method for each instance. Develop a python program to implement the scenario.
7. Experiments on Python Data Series, Data Frames, Binary operations in Data Frames, Missing data and filling values, etc. in Pandas Operations.

EXPERIMENT 4

Perform the following operations:

1. Apply the use of ATTRIB windows command to change the attributes of a file.
2. Create a file xyz.txt and change the ownership of this file to some other user on your machine
3. Create a file hello.txt and make it executable.
4. Create a new user account and home directory called “Duck” and Set the user account “Duck’s” expiry date as 07-07-2020
5. Check the network connectivity of your computer using suitable Linux commands.

EXPERIMENT 5

Perform the following:

1. Create a directory in your machine and make it as a repository and perform the following
    a. Create a text file and add some content into it.
    b. Add the file to your repository
    c. Commit the file to your repository
    d. See the commit details using git log command.
2. Go to your Git repository and perform the following
    a. Do some modifications in your text file. Commit the changes.

Try to revert to your old revision, again do some modifications in your text file and try to discard the changes.

Faculty-in-Charges: Ms. Jeshmol P J
Ms. Bency Wilson
Ms. Ancy C A
Mr. Ajith Jacob

OPEN QUESTIONS

1. Write a Python program to check if a given positive integer is a power of two.
2. Write a Python program to find three numbers from an array such that the sum of there numbers equal to zero
3. Write a Python program to find the single element appears once in a list where very element appears four times except for one.
4. Write a Python program to check a sequence of number is an arithmetic progression or not
5. Write a Python program to get the Hamming numbers upto a given numbers also check whether a given number is an Hamming number.
6. Write a Python program to find majority element in a list.
7. Write a Python program to find the largest palindrome made from the product of a two 4-digit numbers.
8. Write a Python program to find the 1000th prime numbers.
9. Given an unordered list of flights taken by someone, each represented as (origin, destination) pairs, an a starting airport. Write a Python program to compute the person's itinerary. If no such Itinerary exists, return null. If there are multiple possible itineraries, return the lexicographically smallest one. All flights must be used in the itinerary.
10. Write a Python program to remove the parentheses area in a string.

ADVANCED QUESTIONS

1. Write a Python program to find the operating system name, platform and platform release date.
2. Write a Python program to find the number of divisors of a given integer is even or odd.
3. Write a Python program to find the digits which are absent in a given mobile number.
4. Write a Python program to compute they summation of the absolute difference of all distinct pairs in a given array
5. Write a Python program to reverse the digits of a given number and add it to the original, If the sum is not a palindrome repeat this procedure.
6. Write a Python program to set the indentation of the first line.
7. Write a Python program to find the first non-repeating character in a given string.
8. Write a Python program to print all permutations with a given repetition number of characters of a given string.
9. Write a Python program to compute the similarity between two lists.
10. Write a Python program to replace the last element in a list with another list.