SEMESTER 3

PERIOD: JULY 2020 - DECEMBER 2020

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Department of 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.

INDEX

Assignment Schedule

SI No	Subject Code & Name	Faculty in-charge	Week
1	**MAT203 DISCRETE MATHEMATICAL STRUCTURES**	Ms.Radhika Das	WEEK 2
2	**ITT201 DATA STRUCTURES**	Dr.Preetha K G	WEEK 3
3	**ITT203 DIGITAL SYSTEM DESIGN**	Ms.Bency Wilson	WEEK 4
4	**ITT205 PROBLEM SOLVING USING PYTHON**	Dr.Saritha S	WEEK 5
5	**EST200 DESIGN & ENGINEERING**	Ms.Divya James	WEEK 6
6	**MCN201 SUSTAINABLE ENGINEERING**	Dr.Biju Paul	WEEK 7
7	**MAT203 DISCRETE MATHEMATICAL STRUCTURES**	Ms.Radhika Das	WEEK 8
8	**ITT201 DATA STRUCTURES**	Dr.Preetha K G	WEEK 9
9	**ITT203 DIGITAL SYSTEM DESIGN**	Ms.Bency Wilson	WEEK 10
10	**ITT205 PROBLEM SOLVING USING PYTHON**	Dr.Saritha S	WEEK 11
11	**EST200 DESIGN & ENGINEERING**	Ms.Divya James	WEEK 12
12	**MCN201 SUSTAINABLE ENGINEERING**	Dr.Biju Paul	WEEK 13

MAT203 DISCRETE MATHEMATICAL STRUCTURES

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY
COURSE INFORMATION SHEET
MAT 203: DISCRETE MATHEMATICS

PROGRAMME:**COMPUTER SCIENCE AND ENGINEERING**	DEGREE: **BTECH**
COURSE: **DISCRETE MATHEMATICS**	SEMESTER: **III** CREDITS: **4**
COURSE CODE**:** **MAT 203** REGULATION: **2020**	COURSE TYPE: **CORE**
COURSE AREA/DOMAIN: **Logic Development**	CONTACT HOURS: **3+1(Tutorial) hours/Wek**.
CORRESPONDING LAB COURSE CODE (IF ANY):	LAB COURSE NAME:

SYLLABUS:

UNIT	DETAILS	HOURS
I	**Module 1:(Fundamentals of Counting Theory)** **(Relevant topics from sections 1.1, 1.2, 1.3, 1.4, 5.5 , 8.1, 8.2, 8.3** 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
II	**Module- 2 (Fundamentals of Logic)** **(Relevant topics from sections 2.1, 2.2, 2.3, 2.4.)** 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
III	**Module 3 (Relations and Functions)** **(Text 1: Relevant topics from sections 5.1, 5.2, 7.1,7.2,7.3,7.4, 7.6)** 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)** **(Relevant topics from sections 9.1,9.2, 9.4, 10.1, 10.2, 10.3)** Generating Function - Definition and Examples - Calculation techniques, Exponential generating function .First order linear recurrence relations with constant Coefficients-homogeneous nonhomogeneous- Solution. The second-order linear recurrence relation with Constant Coefficients - homogeneous- Nonhomogeneous – Solution.- Three cases, The Nonhomogeneous Recurrence Relation -First order- Second Order	9
V	**Module-5 (Algebraic Structures)** **(Relevant topics from sections 15.1, 15.2, 15.3, 15.4, 15.5)** 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	9

	three symbols-The direct product of two groups. Group Homomorphism- Isomorphism of groups, Cyclic group. Right coset left cosets- Lagrange's Theorem	
	TOTAL HOURS	45

TEXT/REFERENCE BOOKS:

Text Book

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

Reference Books

1) Kenneth H. Rosen, Discrete Mathematics and Its Applications with Combinatorics and Graph Theory, Seventh Edition, MGH, 2011

2) Trembly J.P and Manohar R, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGrawHill Pub. Co. Ltd., New Delhi, 2003.

3) Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, "Discrete Mathematical Structures", Pearson Education Pvt Ltd., New Delhi, 2003

4) Kenneth H.Rosen, "Discrete Mathematics and its Applications", 5/e, Tata Mc Graw Hill Pub.Co.Ltd, New Delhi 2003

5) Richard Johnsonbaugh, "Discrete Mathematics", 5/e, Pearson Education Asia, New Delhi, 2002.

6) Joe L Mott, Abraham Kandel, Theodore P Baker, "Discrete Mathematics for Computer Scientists and Mathematicians", 2/e, Prentice-Hall India, 2009.

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 or counting 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 structures	Solve/Apply (level 3)

	its properties such as monoids and groups	

CO-PO AND CO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C203.1	3	2	3	2	1	_	_	_	_	2	_	1	_	_	_
C203.2	3	2	3	2	3	_	_	_	_	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 complex engineering solutions
CS203.5-PO12	L	Algebraic structures will help life-long learning in the broadest context of 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	

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	Different types of numbers and their properties	Class Assignment	1,3	3

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 **Approved by**

Radhika Das

HOD

COURSE PLAN

No	Topic	No. of Lectures
1	**Fundamentals of Counting Theory9 hours**	
1.1	The Rule of Sum – Extension of Sum Rule	1
1.2	The Rule of Product - Extension of Product Rule - Permutations	1
1.3	Combinations, The Binomial Theorem (Without proof)Combination withrepetition	2
1.4	The Pigeonhole Principle	1
1.5	The principle of Inclusion and Exclusion Theorem (Without Proof) Generalization of theprinciple	2
1.6	Derangements	2
2	**Fundamentals of Logic9 hours**	

2.1	Mathematical logic -Basic Connectives and Truth Table: Statements – Logical Connectives – Tautology - Contradiction	3
2.2	Logical Equivalence: The Laws of Logic- The principle of duality- Substitution Rules	2
2.3	The implication-The Contrapositive, the Converse – the Inverse	1
	Logical Implication : Rules of Inference – Logical Implication The Use of Quantifiers: Open Statement- Quantifier	1

2.4	Logically Equivalent – Contrapositive – The Converse – The Inverse – Logical Implications – Negation	2
3	**Relations and Functions9 hours**	
3.1	Cartesian Product - Binary Relation Function – Domain – Range – One to One Function Image- Restriction	1
3.2	Reachability Relations, Reflexive Relations-Symmetric Relations- Transitive relations - Antisymmetric Relations -	1
3.3	Partial Order relations -Equivalence Relation- Irreflexive Relations.	1
3.4	Zero-one Matrices and directed graphs- Composite Relations- Zero One Matrix-Relation Matrix - Directed Graph - Strongly Connected Directed Graph. Partially ordered Set - HasseDiagram- Maximal-Minimal Element-Least Upperbound- Greatest Lower Bound	2
3.5	Partially ordered Set - Hasse Diagram-Maximal-Minimal Element- Least Upperbound- Greatest Lower Bound	2
3.6	Equivalence Relation - Equivalence Class	1
	Lattice- Dual Lattice – sublattice – Properties of glb and lub – Properties of Lattice - Special Lattice : Complete Lattice – Bounded Lattice – Completed Lattice – Distributive Lattice	1
4	**Generating Functions and Recurrence Relations9 hours**	
4.1	Generating Function - Definition and Examples, Exponential Generating Function.	2
4.2	First Order Linear Recurrence Relations with Constant Coefficients Homogeneous- Nonhomogeneous- Solution	2
4.3	The Second-Order Linear Recurrence Relation with Constant Coefficients -Homogeneous Nonhomogeneous Solution	3

4.4	The Nonhomogeneous Recurrence Relation -First order- Second Order	2

5	**Algebraic Structures 9 hours**	
5.1	Algebraic System-Properties- Homomorphism and Isomorphism	1
5.2	Cyclicmonoid, subsemigroup and submonoid- homomorphism and Isomorphism of Semigroup and Monoids.	2
5.3	ElementaryProperties- Subgroup- symmetric group on three symbols-The direct Product of two Groups	2
5.4	Group Homomorphism- Isomorphism- Cyclic group	2
5.5	Right coset- Left Cosets- Lagrange's Theorem	2

Assignment Questions

1. Suppose there are 8 males and 5 female professors teaching calculus class. How many ways a student can choose a calculus professor?
2. Buick automobiles come in 4 models, 12 colours, 3 engine sizes, and 2 transmission types,
   (a) how many distinct Buicks can be manufactured?
   (b) If one of the available colours is blue, how many different blue Buicks can be manufactured?
3. In a class of 10 students, five are to be chosen and seated in a row for a picture. How many such linear arrangements are possible?
4. How many 4 digit numbers can be formed by using the digit 1to 9 if repetition of digits is not allowed.
5. Find the number of combinations of the four objects a, b, c, d taken three at a time.
6. In how many ways can a gambler draw five cards from a standard deck and get (a) a flush (five cards of the same suit)?(b) four aces? (c) four of a kind? (d) three aces and two jacks? (e) three aces and a pair? (f) a full house (three of a kind and a pair)?(g) three of a kind? (h) two pairs?
7. Determine the coefficient of (i) xyz^2 in $(x+y+z)^4$ (ii) xyz^{-2}in$(x-2y+3z^{-1})^4$.
8. In how many ways can 10 (identical) dimes be distributedamong five children if (a) there are no restrictions? (b) each child gets at least one dime? (c) the oldest child gets at least two times?
9. Show that if eight people are in a room, at least two of them have birthdays that occur on the same day of the week.

10. In how many ways can Troy select nine marbles from a bag of twelve (identical except for colour), where three are red, three blue, three white, and three green?

Tutorial Questions

1. Suppose E is an event of choosing a prime number between 10 and 20 and suppose F is the event of choosing an even number between 10 and 20.How many ways does E or F occur?
2. The board of directors of a pharmaceutical corporation has10 members. An upcoming stockholders' meeting is scheduled to approve a new slate of company officers (chosen from the 10 board members).

   a) How many different slates consisting of a president, vice president, secretary, and treasurer can the board present to the stockholders for their approval?

   b) Three members of the board of directors are physicians appearing on theslate?
3. Over the Internet, data are transmitted in structured blocks of bits called *datagrams.*

   a)In how many ways can the letters in DATAGRAM be arranged?

   b) For the arrangements of part.
4. How many arrangements of the letters in MISSISSIPPI have no consecutive S's?
5. In how many ways can the letters of the word MATHEMATICS be arranged such that the vowels must always come together? In how many ways different arrangements can be made with this letters.
6. Find the coefficient of $x^2y^2z^3$ in the expansion of $(x+y+z)^7$.
7. Determine how many ways 20 coins can be selected from four large containers filled with pennies, nickels, dimes, and quarters. (Each container is filled with only one type of coin.)
8. An auditorium has a seating capacity of 800. How many seats must be occupied to guarantee that at least two people seated in the auditorium have the same first and last initials?
9. Determine the number of positive integers n, $1 < n < 2000$,that are

   a) not divisible by 2, 3, or 5

   b) not divisible by 2, 3, 5, or 7

   c) not divisible by 2, 3, or 5, but are divisible by 7
10. How many derangements are there for $1, 2, 3, 4, 5$?

ITT201 DATA STRUCTURES

COURSE INFORMATION SHEET

PROGRAMME: **INFORMATION TECHNOLOGY**	**DEGREE: BTECH**
COURSE: **DATA STRUCTURES**	SEMESTER: III CREDITS: 4
COURSE CODE: **ITT 201**REGULATION: 2019	COURSE TYPE: **CORE**
COURSE AREA/DOMAIN: PROGRAMMING	CONTACT HOURS: **3+1(Tutorial) hours/Week.**
CORRESPONDING LAB COURSE CODE (IF ANY):**ITL201**	LAB COURSE NAME: DATA STRUCTURES LAB

SYLLABUS:

UNIT	DETAILS	HOURS
I	Data Structures-Introduction and Overview- Arrays, Algorithm/Program Development, Searching and Sorting.	9
II	Linked lists, singly linked list, Doubly linked list, Circular linked list, Applications of linkedlist, Dynamic Memory management.	10
III	Stack, Applications of stacks, Queues, Types of queues	9
IV	Trees, Binary Tree Traversals, Binary tree Applications, Graph, and Graph Applications.	10
V	Hash Tables, Different Hash Functions, Collision Resolution Techniques, closed hashing and Open Hashing (Separate Chaining).	7
	TOTAL HOURS	45

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION

T1	Samanta D., Classic Data Structures, Prentice Hall India, 2/e, 2009.
T2	Ellis horowitz, SartajSahni, Fundamentals of Data structures, GalgotiaBooksource
R2	Horwitz E., S. Sahni and S. Anderson, Fundamentals of Data Structures in C, University Press (India), 2008.
R2	Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication,1983.
R3	Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill, 1995.
R4	Peter Brass, Advanced Data Structures, Cambridge University Press, 2008
R5	Lipschuts S., Theory and Problems of Data Structures, Schaum's Series, 1986.
	Wirth N., Algorithms + Data Structures = Programs, Prentice Hall, 2004.
	Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI, 1987.
	Martin Barrett, Clifford Wagner, And Unix: Tools For Software Design, John Wiley, 2008 reprint.

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
	Programming in C	C Programming	S2

COURSE OBJECTIVES:

1	To impart a thorough understanding of linear data structures such as stacks, queues and their applications.
2	To impart a thorough understanding of non-linear data structures such as trees, graphs and their applications.
3	To impart familiarity with various sorting, searching and hashing techniques and their performance comparison.
4	To impart a basic understanding of memory management.

COURSE OUTCOMES:

After the completion of the course the student will be able to

CO_No	Course Outcome (CO)	**Bloom's Category**

CO1	**Summarize** different categories of data Structures	Level 2: Understand
CO2	**Identify different** parameters to analyze the performance of an algorithm.	Level 3: Apply
CO3	**Explain** the significance of dynamic memory management Techniques	Level 2: Understand
CO4	**Design** algorithms to perform operations withLinear and Nonlinear data structures	Level 3: Apply
CO5	**Illustrate** various technique to for searching, Sorting and hashing	Level 2: Understand
CO6	**Choose** appropriate data structures to solve realworld problems efficiently.	Level 3: 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	P0 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	-	-	1	-	-	-	-	-	1	2	1	1
CO2	3	2	2	2	1	1	-	-	-	-	-	1	3	-	-
CO3	3	3	3	2	1	1	-	-	-	-	-	1	3	1	-
CO4	3	3	3	2	1	1	-	-	-	-	-	1	2	-	1
CO5	3	2	2	1	1	-	-	-	-	-	-	1	1	-	-
CO6	3	3	3	2	1	1	-	-	-	-	-	1			

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	LOW/MEDIUM/HIGH	Justification
CO1-PO1	H	The knowledge in programming methodologies helps in designing solutions for complex engineering problems.
CO2-PO2	M	The knowledge of asymptotic notations helps in analysis of performance of solutions to complex problems
CO3-PO3	M	The knowledge in programming methodologies and asymptotic notations help in designing solutions and analyzing its complexity.
CO-PO6	L	The knowledge in algorithms help in designing solutions of the societal problems.
CO1-PSO1	M	This knowledge helps to design good and efficient algorithms.
CO1-PSO2	L	The knowledge in data structure will help to design various less complex networking algorithms
CO1-PSO3	L	These concepts are fundamental to CS and can be used in research and other innovative ideas.
CO2-PO1	H	The knowledge of arrays, linked lists, stacks and queues can be applied to solve complex engineering problems.

CO2-PO2	M	The knowledge of arrays, linked lists, stacks and queues can be applied to design solutions to complex engineering problems.
CO2-PO3	M	The knowledge in and asymptotic notations help in designing solutions and analyzing its complexity
CO2-PO4	M	Knowledge in algorithms complexity leads to the development of better solutions
CO2-PO5	1	These concepts are fundamental to CS and can be used for developing solutions of industrial problems.
CO2-PO6	1	Various algorithms can be developed for societal needs
CO2-PO12	1	The knowledge in algorithm help to develop solutions of different problems in the life long
CO2-PSO1	H	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.
CO3-PO1	H	The knowledge of non linear data structures like trees and graphs can be applied to solve complex engineering problems.
CO3-PO3	H	This knowledge can be used to design efficient solutions to complex problems.
CO3-PO4	H	This knowledge helps in representation, analysis and interpretation of data to provide valid conclusions.
CO3-PSO1	H	The knowledge of non linear data structures like trees and graphs can be applied to design solutions to complex engineering problems.
CO3-PSO2	L	This knowledge helps in designing efficient algorithms using appropriate data structure.
CO4-PO1	L	This basic knowledge of sorting and searching can be used in solutions to complex engineering problems.
CO4-PO3	M	This basic knowledge of sorting and searching can be used in designing solutions to complex engineering problems.
CO4-PO4	H	This concept is fundamental in conducting investigations and interpretations of data.
CO4-PSO1	M	This basic knowledge of sorting and searching can be used in designing solutions to complex multidisciplinary engineering problems.
CO4-PSO3	L	The concept of sorting and searching are fundamental to the CS discipline and can be used research and other innovative ideas.
CO5-PO1	L	The knowledge of various hashing techniques can be applied in designing solutions to complex engineering problems.

CO5-PSO1	L	The knowledge of various hashing techniques can be applied in designing solutions to complex multidisciplinary engineering problems.
CO6-PO1	H	Apply the knowledge in the data structure in the real world problems
CO2-PO2	M	Finding solutions to actual problems with less complexity
CO3-PO3	M	Design and implementations of various algorithms for real world problems
CO3-PO4	L	Analysing the complexity of different algorithms to solve real life problems

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

SNO	DESCRIPTION	PROPOSED ACTIONS
1	Sparse Matrix Operations	Learning Material provided.
2	Towers of Hanoi Problem(Example of recursion)	Learning Material provided.

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

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

1	AVL Trees

WEB SOURCE REFERENCES:

1	http://nptel.ac.in/courses/106103069

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 **Approved by**

Dr.Preetha K G **HOD, IT**

Course Plan – Aug 2020

Sl.No	Day	Module	Topics
1	Day 1	1	Introduction to Algorithms
2	Day 2	1	Stepwise refinement techniques
3	Day 3	1	Programming Style- Modular Programmimg and Structural Programming
4	Day 4	1	Recursive Algorithms
5	Day 5	1	Analysis of algorithms
6	Day 6	1	Analysis of algorithms
7	Day 7	1	Asymptotic Notations
8	Day 8	1	Tutorial
9	Day 9	3	Stack Using Array
10	Day 10	3	Application of Stack- Infix to postfix
11	Day 11	3	Postfix Evaluation
12	Day 12	3	Queue Using Array, Circular queue
13	Day 13	3	Deque
14	Day 14	3	Tutorial
15	Day 15	2	Linked lists- Singly Linked list
16	Day 16	2	Doubly Linked list
17	Day 17	2	Circular list
18	Day 18	2	Application of LL- Polynomial Addition
19	Day 19	5	Tutorial
20	Day 20		Searching Techniques- Linear , Binary
21	Day 21	5	Sorting Techniques- Selection Sort
22	Day 22	5	Bubble Sort
23	Day 23	5	Insertion sort
24	Day 24	5	Quick sort
25	Day 25	5	Merge sort
26	Day 26	5	Tutorial
27	Day 27	4	Tree- Binary Tree- Terminologies
28	Day 28	4	Tree Implementation

29	Day 29	4	Tree Traversals
30	Day 30	4	Binary Search Tree
31	Day 31	4	Operations on BST
32	Day 32	4	Tutorial
33	Day 33	5	Representation of Graph
34	Day 34	5	BFS
35	Day 35	5	DFS
36	Day 36	3	Memory Management
37	Day 37	3	Memory allocation schemes
38	Day 38	6	Hash Tables
39	Day 39	6	Midsquare, Division
40	Day 40	6	Folding, Digit Analysis
41	Day 41	6	Collision Resolution
42	Day 42	6	Overflow Handling Techniques
43	Day 43	6	Tutorial
44	Day 44		University Question Paper Discussion
45	Day 45		University Question Paper Discussion

ASSIGNMENT QUESTIONS

Assignment 1

1. Write about Sparse Matrix with suitable examples. Write an algorithm for sparse matrix multiplication

Assignment 2

1. Write an algorithm to make the copy of a binary tree.
2. Write an algorithm to create the mirror image of the tree
3. Write an algorithm 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

TUTORIAL QUESTIONS

1. Write an algorithm to perform the following operations on array
   - Display array element
   - Sort the array elements
   - Search an element from array
   - Replace an element with other element
2. Write an algorithm to perform the following operations on String
   - Length of string
   - String Concatenation
   - Sting Reverse
   - String Copy
4. Write the postfix from the following expressions:

- A*B*C*D
- –A+B-C+D
- (A+B)*D+E/(F+A*D)+C

5. The prefix form of the expression (A+B) ^(C-D) is----------

ITT203 DIGITAL SYSTEM DESIGN

COURSE INFORMATION SHEET

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of understanding the basic digital logic design and implementation. All students of computing should acquire some understanding and appreciation of a computer system's functional components, their characteristics, their performance, and their interactions.

Prerequisite: NIL

Course Outcomes: After the completion of the course, the student will be able

CO No.	Course Outcome(CO)	Bloom's Category
CO 1	To perform base conversion and arithmetic operations in various 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

Mapping of course outcomes with program outcomes

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	3			3	-	-	-	-	-	-	-	2
CO 2	3	3	3	3	-	-	-	-	-	-	-	2
CO 3	3	3	3	3	-	2	-	1	-	-	-	2
CO 4	3			3	-	-	-	-	-	-	-	2
CO 5	3	3	3	3	2	2	-	1	-	-	-	2

3/2/1: high/medium/low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	10	5	20
Understand	15	10	20
Apply	10	10	25
Analyse	10	10	15
Evaluate	5	10	10

Create		5	10

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	4 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks
Continuous Assessment Test (2 numbers) : 25 marks
Assignment/Quiz/Course project : 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1): To understand the basic concepts of Number systems

1. Convert the given number from decimal number system to binary, octal, and hexadecimal number system.
2. Perform Arithmetic operations on different number system.
3. Represent the different coding schemes.

Course Outcome 2 (CO2): To design digital circuits using simplified Boolean functions

1. Simplify the given expression using Postulates of Boolean algebra.
2. Convert a given expression to standard and canonical forms.
3. Simplify the given expression using Karnaugh Map or Quine –McClusky minimization technique.

Course Outcome 3 (CO3): To analyze and design combinational circuits

1. Analyse a given circuit and explain the results obtained by the circuit.
2. Design a Ccarry look ahead adder.
3. Design a four-bit magnitude comparator.

Course Outcome 4 (CO4): To understand the basics of sequential circuits

1. Understand the functioning of Latches and Flip Flops.
2. Design Master-Slave Flip Flops.
3. Understand the basics of different types of Flip Flops.

Course Outcome 5(CO5): To analyze and design synchronous and asynchronous sequential circuits

1. Analyse a given circuit and explain the results obtained by the circuit.
2. Implement a serial adder using a shift register.
3. Design and construct a 4-bit ring counter with only one flip-flop is clear at any particular time and all other flip-flops are set. Give its timing diagram.
4. Using an example, show the Race-Free State Assignment in an asynchronous sequential circuit.

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 compliment. Arithmetic Operations – Addition, Subtraction, Multiplication, Division of Binary numbers, Booths 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- Universal Gates.
Module 3: COMBINATIONAL LOGIC (9 Hours)
Combinational Circuits – Analysis and Design Procedures - Binary Adder-Sub tractor (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: COUTERS 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 Books

1. Mano M. M. and Michael D. Ciletti, *Digital Design*, 4/e, Pearson Education, 2013.
2. Thomas L. Floyd, *Digital Fundamentals*, 11th Edition, Pearson Education, 2015.
3. N. N. Biswas, *"Minimization of Boolean Functions,"* in *IEEE Transactions on Computers*, vol. C-20, no. 8, pp. 925-929, Aug. 1971. doi: 10.1109/T-C.1971.223373

Reference Books

1. Charles H Roth ,Jr, Lizy Kurian John, *Digital System Design using VHDL*,2/e, Cengage Learning
2. Mano M. M. and Michael D. Ciletti, Digital Design with an Introduction to the Verilog HDL, 5/e, Pearson Education, 2013.
3. Tokheim R. L., Digital Electronics Principles and Applications, 7/e, Tata McGraw Hill, 2007.
4. Rajaraman V. and T. Radhakrishnan, *An Introduction to Digital Computer Design*, 5/e, Prentice Hall India Private Limited, 2012.
5. Leach D, Malvino A P, Saha G, *Digital Principles and Applications*, 8/e, McGraw Hill Education, 2015.
6. M. Morris Mano, *Computer System Architecture*, 3/e, Pearson Education, 2007.
7. Harris D. M. and, S. L. Harris, *Digital Design and Computer Architecture*, 2/e, Morgan Kaufmann Publishers, 2013

COURSE PLAN

No	Topic	No. of Lectures
1	**NUMBER SYSTEM**	**9 Hours**
1.1	Number Systems – Decimal, Binary, Octal, Hexadecimal - conversion from one system to another – Representation of	3 hours

	negative numbers using 2's compliment.	
1.2	Arithmetic Operations – Addition, Subtraction, Multiplication, Division of Binary numbers, Booths algorithm for multiplication, representation of negative numbers, Representation of floating point 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	**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	**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 – Multiplexers	3 Hours
3.3	Parity Generator– Multiplexers – DE multiplexers – Implementation of Boolean functions using MUX.	3 Hours
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	**COUTERS 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 Hour

ASSIGNMENT QUESTIONS

Assignment -I

1. Describe the ASCII and EBCDIC code in detail.

Assignment -II

1. Draw 5 and 6 variable K map and mark the minterms in it.

Assignment - III

1. Implement the following Boolean functions using simple AND, OR and NOT logic gates (do not simplify the functions):

(a) $F = A\overline{B} + A\overline{B}C + CD$ (b) $F = \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}$

(c) $F = \overline{A} + AB\overline{C} + A\overline{B}C$ (d) $F = X\overline{Y}\,\overline{Z}(W + Y\overline{Z}) + \overline{Z}W + \overline{X}\,\overline{Y}$

2. Reduce the following expressions, using Boolean algebraic methods. State the relevant law or postulate used at each step.

(a) $X.X.Y$ (b) $X + \overline{X} + Y$ (c) $(X + \overline{X}).B$
(d) $B + B.A$ (e) $Y.(\overline{Y}+X)$ (f) $(A+B).(A+C)$

3. Using **only** the theory of Boolean algebra and algebraic manipulation, simplify the following Boolean expressions to a minimum number of literals:

(a) $A + BC + (B + \overline{C})(\overline{B} + A)$ (b) $\overline{A} + AB\overline{C} + B$
(c) $A\overline{B}\,\overline{D}(A + B\overline{D}) + \overline{D}A + \overline{B}\,\overline{C}$ (d) $(\overline{XY}) + X\,\overline{Y} + Z(X + YZ) + \overline{X}\,\overline{Y}$

4. Write the following functions in shorthand "π" product of maxterms form:

(a) $F(A,B,C) = (\overline{A} + B + \overline{C})(A + \overline{B} + \overline{C})(A + \overline{B} + C)$
(b) $G(W,X,Y,Z) = (\overline{W}+X+Y+\overline{Z})(\overline{W}+\overline{X}+\overline{Y}+\overline{Z})(W+\overline{X}+Y+Z)$
(c) $H(A,B,C,D) = (A+C)(B+C)(A+\overline{B})(A+D)$

5. Write the following functions in shorthand "$\sum$" sum of minterms form:

(a) $F(A,B,C) = \overline{A}\,BC + A\overline{B}C + \overline{A}B\overline{C}$
(b) $G(W,X,Y,Z) = \overline{W}\,\overline{X}YZ + WX\,\overline{Y}Z + \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z} + WXY\overline{Z} + \overline{W}\,\overline{X}\,\overline{Y}Z + WXYZ$
(c) $H(A,B,C) = \overline{A}\,B + \overline{B}C + \overline{A}C$

TUTORIAL QUESTIONS

1. Convert the following binary numbers to decimal: 101110; 1110101; and binary
2. Convert the following numbers with the indicated bases to decimal: (12121)3; (4310)s; (50)7,; and (198)12.
3. Convert the following decimal numbers to binary: 1231; 673; and 1998.
4. Convert the following decimal numbers to the bases indicated.
   a. 7562 to octal
   b. 1938 to hexadecimal
   c. 175 to binary
5. Convert the hexadecimal number F3A7C2 to binary and octal.
6. What is the radix of the numbers if the solution to the quadratic equation r2 - lOr + 31 = 0 is x = 5 and r = 8?
7. Show the value of all bits of a 12-bit register that hold the number equivalent to decimal 215 in (a) binary; (b) bi.nary-coded octal; (c) binary coed hexadecimal; (d) binary coded decimal (BCD).
8. Write your name in ASCII using an 8-bil code with the leftmost bit always
   a. Include a space between names and a period after a middle initial.
9. Obtain the 9's complement of the following eight-digit decimal numbers: 12349876; 00980100; 90009951; and 00000000.
10. Obtain the 10's complement of the following six-digit decimal numbers: 123900; 090657; 100000; and 000000.
11. Obtain the l's and 2's complements of the following eight-digit binary numbers: 10101 1 10; 10000001; 10000000; 00000001; and 00000000.
12. Perform the subtraction with the following unsigned decimal numbers by taking the 10's complement of the subtrahend.
    a. 5250 – 1321 c.. 1753 - 8640
    b. 20 - 100 d. 1200 – 250

13. Perform the subtraction with the following unsigned binary numbers by taking the 2's complement of the subtrahend.
    a. 11010 – 10000 c.. 1 1010 - 1101
    b. 100 - 1 1 0000 d. 1010100 – 1010100.

14. Perform the arithmetic operations (+42) + (- 13) and (-42) - (- 13) in binary using signed-2's complement representation for negative numbers.

15. Perform the arithmetic operations (+70) + (+80) and (-70) + (-80) with binary numbers in signed-2's complement representation. Use eight bits to accommodate each number together with its sign. Show that overflow occurs in both cases, that the last two carries are unequal, and that there is a sign reversal.

16. DeteMine by means Via truth table the validily of DeMorgan's theomn for three variables: (ABC)' • A' + B' + C'.

17. Simplify the following expressions using Boolean algebra.

a. A + AB
b. AB + AB'
c. A' BC + AC
d. A'B +ABC' + ABC

18. Simplify the following expressions using Boo lean algebra.

a. AB + A(CD + CD')
b. (BC' + A'D) (AB' + CD')

19. Using DeMcngan's thoomn, show that

a. (A + B)'(A' + B')' = 0
b. A + A'B + A'B' =1

20. Given the Boolean function F =x'y + xyz':Derive an algebraic expression for the complement f'.
a.Show that F • F' a=0.
b.Show that F + F' = I.

21. Given the Boolean function

F =.xy'z + z'y'z + xyz

ITT205 PROBLEM SOLVING USING PYTHON

COURSE INFORMATION SHEET

PROGRAMME: INFORMATION TECHNOLOGY	DEGREE: BTECH
COURSE: PROBLEM SOLVING USING PYTHON	SEMESTER: III CREDITS: 4
COURSE CODE: ITT 205 REGULATION: 2019	COURSE TYPE: CORE
COURSE AREA/DOMAIN: PROGRAMMING	CONTACT HOURS: 3+1(Tutorial) hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY):ITL 203	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	**Python Object Oriented Programming:** Introduction to classes and objects - class definition, attributes, instances, sameness, instances as arguments and return values. Constructor, class attributes and destructors, Inheritance.	9
	TOTAL HOURS	45

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION

T T R R R R R R	Downey, A. et al., How to think like a Computer Scientist: Learning with Python, John Wiley, 2015 Mark Lutz,"Learning Python: Powerful Object-Oriented Programming" , O'Reilly Media Inc.,5th,2013 Kenneth A. Lambert, B. L. Juneja, "Fundamentals of Python", Cengage Lerning India Pvt. Ltd., 2015. S.A.Kulkarni, "Problem Solving and PYTHON Programming", 2nd edition, Yes Dee Publishing Pvt Ltd, 2018 Mark Summerfield,"Programming in Python 3: A Complete Introduction to the Python Language", Pearson Education, 2nd,2018 Yashavant Kanetkar ,Aditya Kanetkar ,"Let Us Python ",BPB Publications, 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 solutions	Apply

CO-PO AND CO-PSO 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**

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

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

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

WEB SOURCE REFERENCES:

1	https://archive.org/details/MIT6.OOSCS 11
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
Dr. Saritha S

Approved by
HOD

COURSE PLAN

No	Topic	No. of Lectures
1	**Introduction To Python: 9 hours**	
1.1	Understanding Python-identifiers, variables, keywords, expressions and statements.	2
1.2	Evaluation of expressions, Operators and operands, operator precedence, indentation	1
1.3	Python Program Flow Control:Decision making- if, if..else, elif.	2

1.4	Loops - for, while, for...else, while...else	2
1.5	Control statements using pass, continue, break.	2
2	**Strings and lists: 9 hours**	
2.1	String traversal, string slices and comparison with examples	1
2.2	The string module, character classification.	1
2.3	List- List values, accessing elements, list membership, Lists and for loops, List operations, List slices, List deletion	2
2.4	Matrices	1
2.5	Tuples- mutability and tuples, tuple assignment, tuples as return values, Tuple operations.	2
2.6	Dictionaries – operations and methods.	2
3	**Python Functions, Modules And Packages: 9 hours**	
3.1	Function definition, calling functions, parameters and arguments, the return statement.	1
3.2	Type conversion and coercion, composition of functions	1
3.3	Lambda function, mathematical functions	1
3.4	user-defined functions	1
3.5	Recursion	1
3.6	Modules -Built-in modules	1
3.7	Creating modules, import statement.	1
3.8	Packages in Python - importing modules from a package.	2
4	**Python Files and exceptions: 9 hours**	
4.1	Python file handling, open, write, read text files	4
4.2	Writing variables	1
4.3	Directories in Python	1
4.4	Pickling	1
4.5	Exception Handling.	2
5	**Python Object Oriented Programming: 9 hours**	
5.1	Introduce classes and objects	1
5.2	Class definition, attributes, instances, sameness	1

5.3	Instances as arguments and return values.	1
5.4	Constructor	2
5.5	Class attributes and destructors	2
5.6	Inheritance	2

ASSIGNMENT I

1. Functions and methods in strings
2. Functions and methods in tuples
3. Functions and methods in lists
4. Functions and methods in dictionaries

ASSIGNMENT II

1. Making Modules in Python
2. Packages in Python
3. Libraries in Python

ASSIGNMENT III

MICRO- PROJECTS ON

a) Dice Rolling Simulator
b) Game Board Printing
c) Password Generator

d) Binary Adder

e) Calculator

f) ATM Machine

g) Address Book for your Class

h) Student Performance Analyzer

i) Game 1 – Rock, Papers and Scissors

j) Game 2 – Tic, Tac, Toe

TUTORIAL I

1. What is the output when the following statement is executed?
   ```
   >>> “a” + “bc”
   ```

2. What is the output when the following statement is executed?
   ```
   >>> “abcd”[2 :]
   ```

3. What is the output when the following statement is executed?
   ```
   >>> str1=”hello”
   >>> str1[-1]
   ```

4. What is the output when the following statement is executed?
   ```
   >>> str1=”helloworld”
   >>>str1[:: -1]
   ```

5. What is the output when the following statement is executed?
   ```
   >>> example = “snow world”
   >>>example [3]=’s’
   >>> print example
   ```

6. What is the output when the following statement is executed?
   ```
   >>> example=”hello”
   >>> example.count(l)
   ```

7. What is the output when the following statement is executed?
   ```
   >>> example =”helle”
   >>> example.find(“e”)
   ```

8. What is the output when the following statement is executed?
   ```
   >>> “hello”+1+2+3
   ```


9. What is the output when the following statement is executed?
   ```
   >>> print (“abcDEF”.capitalize())
   ```

10. What is the output when the following statement is executed?
    ```
    >>> print(“xyyzxyzxzyy”.count(‘yy’,2))
    ```

TUTORIAL II

1. What is the output when the following statement is executed?

```
i = 1
while True:
        if i%2==0:
                break
        print i
i+=2
```

2. What is the output when the following statement is executed?

```
i=2
        while True:
                if i%3==0:
                        break
                print i
        i+=2
```

3. What is the output when the following statement is executed?

```
i=0
while i <5:
        print i
        i+=1
        if i==3:
                break
else :
        print 'x'
```

4. What is the output when the following statement is executed?

```
i=0
while i <5:
        print i
        i+=1
        if i==3:
                break
else :
        print 'x'
```

5. What is the output when the following statement is executed?

```
X="abcdef"
i="a"
while i in X:
        print i
```

6. What is the output when the following statement is executed?

```
X="abcdef"
i="a"
while i in X:
        X=X[1:]
        print i
```


7. What is the output when the following statement is executed?

```
X="abcdef"
i="a"
while i in X[1:]:
        print i
```

8. What is the output when the following statement is executed?

```
X='abcd'
for i in range(len(X)) :
        x='a'
        print x
```

9. What is the output when the following statement is executed?

```
for i in range(2.0) :
        print i
```

10. What is the output when the following statement is executed?

```
X='abcd'
for i in range((len(X)) :
        print X
        X='a'
```

TUTORIAL III

11. What is the output when the following statement is executed?

```
for i in range(5) :
        if i == 5 :
                break
        else  :
                print i
else :
        print("here")
```

12. Identify all the invalid variable names from the following. Also give the reason
    (a) 2sum
    (b) _sum@
    (c) for
    (d) fori

13. What will be the output of the following expressions
    (a) 2*3**2
    (b) 2**3**2

14. What is the output when the following statement is executed?

```
>>> for number in range (30,20,-2):
        number=number+5
        print number
```

15. Which of the following are not valid identifiers in Python? Justify your answer
    a) num_1
    b) 2md_large
    c) num1
    d) attendance%

16. Let x = 2**4/8/2 is an expression in Python. What will be the value of x?

17. print 4 + 2 ** 3 * 6

18. Evaluate the expression
    (a) len(range(4,5)) – 1
    (b) 4 **(3.0/2) = 8

(4 *4 *4)^0.5 = (64)^0.5 = (8*8)^0.5= sqrt(8*8)=8

(c) "a"+"b"*2 - abb

19. What will be the output of the given code?

```
>>> string=”Hello COMPUTER!”
>>>len(string)
```

20. During the slicing operation, if the second index is smaller than the first index, what is the output?

EST200 DESIGN & ENGINEERING

COURSE INFORMATION SHEET

PROGRAMME: INFORMATION TECHNOLOGY	DEGREE: BTECH
COURSE: DESIGN AND ENGINEERING	SEMESTER: III CREDITS: 2
COURSE CODE: EST 200 REGULATION: 2019	COURSE TYPE: CORE
COURSE AREA/DOMAIN: ENGINEERING	CONTACT HOURS: 2 LECTURE HOURS/WEEK
CORRESPONDING LAB COURSE CODE (IF ANY): NIL	LAB COURSE NAME: NIL

SYLLABUS:

UNIT	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	5
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.	5
III	**Design Communication** (Languages of Engineering Design):-Communicating Designs Graphically, Communicating Designs Orally and in Writing. Mathematical Modeling In Design, Prototyping and Proofing the Design.	5
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.	5
V	**Expediency, Economics and Environment in Design Engineering:-**Design for Production, Use, and Sustainability. Engineering Economics in Design. Design Rights. Ethics in Design	5
	TOTAL HOURS	25

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION

T	YousefHaik, SangarappillaiSivaloganathan, Tamer M. Shahin, Engineering Design Process, Cengage Learning 2003, Third Edition, ISBN-10: 9781305253285,
T	Voland, G., Engineering by Design, Pearson India 2014, Second Edition, ISBN 9332535051
R	Philip Kosky, Robert Balmer, William Keat, George Wise, Exploring Engineering, Fourth Edition: An Introduction to Engineering and Design, Academic Press 2015, 4th Edition, ISBN: 9780128012420.
R	Clive L. Dym, Engineering Design: A Project-Based Introduction, John Wiley & Sons, New York 2009, Fourth Edition, ISBN: 978-1-118-32458-5
R	Nigel Cross, Design Thinking: Understanding How Designers Think and Work, Berg Publishers 2011, First Edition, ISBN: 978-1847886361
R	Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., Engineering Design: A Systematic Approach, Springer 2007, Third Edition, ISBN 978-1-84628-319-2

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
	NIL		

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:

CO No.	Course Outcomes	Bloom's Category
CO 1	Explain the different concepts and principles involved in design engineering.	Understand
CO 2	Apply design thinking while learning and practicing engineering.	Apply
CO 3	Develop innovative, reliable, sustainable and economically viable designs incorporating knowledge in engineering.	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	PSO1	PSO2	PSO3
CO 1	2	1					1			1			2		
CO 2		2				1		1				2		1	2
CO 3			2			1	1		2	2		1		2	1

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

SNO	DESCRIPTION	PROPOSED ACTIONS
1	UML Diagrams	Special Lecture

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

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

1	Familiarisation of StarUML http://staruml.io/

WEB SOURCE REFERENCES:

1	https://www.coursera.org/learn/uva-darden-design-thinking-innovation
2	https://www.coursera.org/specializations/software-design-architecture

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

Divya James
(Faculty)

Approved by

Dr. Neeba EA
(HOD)

COURSE PLAN

No	Topic	No. of Lectures
1	**Design Process:** **5 hours**	
1.1	Introduction to Design and Engineering Design	1
1.2	Defining a Design Process-: Detailing Customer Requirements.	1

1.3	Defining a Design Process-: Setting Design Objectives, Identifying Constraints, Establishing Functions.	1
1.4	Defining a Design Process-: Generating Design Alternatives and Choosing a Design	1
1.5	Case Studies:- Stages of Design Process.	1
2	**Design Thinking Approach: 5 hours**	
2.1	Introduction to Design Thinking	1
2.2	Iterative Design Thinking Process Stages: Empathize, Define, Ideate, Prototype and Test.	1
2.3	Design Thinking as Divergent-Convergent Questioning.	1
2.4	Design Thinking in a Team Environment.	1
2.5	Case Studies: Design Thinking Approach.	1
3	**Design Communication: 5 hours**	
3.1	Communicating Designs Graphically.	1
3.2	Communicating Designs Orally and in Writing	1
3.3	Mathematical Modelling in Design.	1
3.4	Prototyping and Proofing the Design	1
3.5	Case Studies: Communicating Designs Graphically.	1
4	**Design Engineering Concepts: 5 hours**	
4.1	Project-based Learning and Problem-based Learning in Design	1
4.2	Modular Design and Life Cycle Design Approaches.	1
4.3	Application of Bio-mimicry, Aesthetics and Ergonomics in Design.	1
4.4	Value Engineering, Concurrent Engineering, and Reverse Engineering in Design	1
4.5	Case Studies: Bio-mimicry based Designs.	1
5	**Expediency, Economics and Environment in Design Engineering: 5 hours**	
5.1	Design for Production, Use, and Sustainability.	1 45
5.2	Engineering Economics in Design.	1
5.3	Design Rights	1

5.4	Ethics in Design.	1
5.5	Case Studies: Design for Production, Use, and Sustainability.	1

ASSIGNMENT I

Develop the following artefacts from different stages of design process:

- Need/Problem Statement
- Prioritized list of requirements
- Objective Tree
- List of Design Constraints
- Functional Structure
- Morphological Chart
- Decision Matrix & Selection of Best Solution(s)
- 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 II

4. Design Thinking Approach
5. Communicating Designs Graphically
6. Bio-mimicry based Designs

MCN201 SUSTAINABLE ENGINEERING

CODE		CATEGORY	L	T	P	CREDIT
MCN201	SUSTAINABLE ENGINEERING		2	0	0	NIL

Preamble: Objective of this course is 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.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the relevance and the concept of sustainability and the global initiatives in this direction
CO 2	Explain the different types of environmental pollution problems and their sustainable solutions
CO 3	Discuss the environmental regulations and standards
CO 4	Outline the concepts related to conventional and non-conventional energy
CO 5	Demonstrate the broad perspective of sustainable practices by utilizing engineering knowledge and principles

Mapping of course outcomes with program outcomes

	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						2	3					2
CO 2						2	3					2
CO 3						2	3					2
CO 4						2	3					2
CO 5						2	3					2

Assessment

Pattern Mark

distribution

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	20	20	40
Understand	20	20	40
Apply	10	10	20
Analyse			
Evaluate			
Create			

Continuous Internal Evaluation Pattern:

Attendance : 10 marks
Continuous Assessment Test (2 numbers) : 25 marks
Assignment/Quiz/Course project : 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B

contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

1

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Course Level Assessment Questions

Course Outcome 1 (CO1): Understand the relevance and the concept of sustainability and the global initiatives in this direction

1. Explain with an example a technology that has contributed positively to sustainable development.
2. Write a note on Millennium Development Goals.

Course Outcome 2 (CO2): Explain the different types of environmental pollution problems and their sustainable solutions

1. Explain the 3R concept in solid waste management?
2. Write a note on any one environmental pollution problem and suggest a sustainable solution.
3. In the absence of green house effect the surface temperature of earth would not have been suitable for survival of life on earth. Comment on this statement.

Course Outcome 3(CO3): Discuss the environmental regulations and standards

1. Illustrate Life Cycle Analysis with an example of your choice.
2. "Nature is the most successful designer and the most brilliant engineer that has ever evolved". Discuss.

Course Outcome 4 (CO4): Outline the concepts related to conventional and non-conventional energy

1. Suggest a sustainable system to generate hot water in a residential building in tropical climate.
2. Enumerate the impacts of biomass energy on the environment.

Course Outcome 5 (CO5): Demonstrate the broad perspective of sustainable practices by utilizing engineering knowledge and principles

1. Suggest suitable measures to make the conveyance facilities used by your institution sustainable.

Model Question paper

Part A

(Answer all questions. Each question carries 3 marks each)

1. Define sustainable development.
2. Write a short note on Millennium Development Goals.
3. Describe carbon credit.
4. Give an account of climate change and its effect on environment.
5. Describe biomimicry? Give two examples.
6. Explain the basic concept of Life Cycle Assessment. 51
7. Name three renewable energy sources.

8. Mention some of the disadvantages of wind energy.

9. Enlist some of the features of sustainable habitat.

10. Explain green engineering.

Part B

(Answer one question from each module. Each question carries 14 marks)

11. Discuss the evolution of the concept of sustainability. Comment on its relevance in the modern world.

OR

12. Explain Clean Development Mechanism.

13. Explain the common sources of water pollution and its harmful effects.

OR

14. Give an account of solid waste management in cities.

15. Explain the different steps involved in the conduct of Environmental Impact Assessment.

OR

16. Suggest some methods to create public awareness on environmental issues.

17. Comment on the statement, “Almost all energy that man uses comes from the Sun”.

OR

18. Write notes on:

    a. Land degradation due to water logging.

    b. Over exploitation of water.

19. Discuss the elements related to sustainable urbanisation.

OR

20. Discuss any three methods by which you can increase energy efficiency in buildings.

Syllabus

Sustainability- need and concept, technology and sustainable development-Natural resources and their pollution, Carbon credits, Zero waste concept. Life Cycle Analysis, Environmental Impact Assessment studies, Sustainable habitat, Green buildings, green materials, Energy, Conventional and renewable sources, Sustainable urbanization, Industrial Ecology.

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.

Reference Books

1. Allen, D. T. and Shonnard, D. R., Sustainability Engineering: Concepts, Design and Case Studies, Prentice Hall.
2. Bradley. A.S; Adebayo,A.O., Maria, P. Engineering applications in sustainable design and development, Cengage learning
3. Environment Impact Assessment Guidelines, Notification of Government of India, 2006
4. Mackenthun, K.M., Basic Concepts in Environmental Management, Lewis Publication, London, 1998
5. ECBC Code 2007, Bureau of Energy Efficiency, New Delhi Bureau of Energy Efficiency Publications-Rating System, TERI Publications - GRIHA Rating System
6. Ni bin Chang, Systems Analysis for Sustainable Engineering: Theory and Applications, McGraw-Hill Professional.
7. Twidell, J. W. and Weir, A. D., Renewable Energy Resources, English Language Book Society (ELBS).
8. Purohit, S. S., Green Technology - An approach for sustainable environment, Agrobios Publication

Course Contents and Lecture Schedule

No	Topic
1	Introduction, concept, evolution of the concept
2	Social, environmental and economic sustainability concepts
1.3	Sustainable development, Nexus between Technology and Sustainable development
1.4	Millennium Development Goals (MDGs) and Sustainable Development Goals (SDGs)
1.5	Clean Development Mechanism (CDM)
2	
2.1	Air Pollution and its effects
2.2	Water pollution and its sources
2.3	Zero waste concept and 3 R concepts in solid waste management
2.4	Greenhouse effect, Global warming, Climate change, Ozone layer depletion
2.5	Carbon credits, carbon trading and carbon foot print.
2.6	Legal provisions for environmental protection.
3	
3.1	Environmental management standards
3.2	ISO 14001:2015 frame work and benefits
3.3	Scope and Goal of Life Cycle Analysis (LCA)
3.4	Circular economy, Bio-mimicking
3.5	Environment Impact Assessment (EIA)
3.6	Industrial Ecology, Industrial Symbiosis
4	
4.1	Basic concepts of Conventional and non-conventional energy
4.2	General idea about solar energy, Fuel cells
4.3	Wind energy, Small hydro plants, bio-fuels
4.4	Energy derived from oceans and Geothermal energy
5	
5.1	Basic concept of sustainable habitat
5.2	Methods for increasing energy efficiency of buildings
5.3	Green Engineering
5.4	Sustainable Urbanisation, Sustainable cities, Sustainable transport

Course Contents and Lecture Schedule

No	Topic
1	Introduction, concept, evolution of the concept
2	Social, environmental and economic sustainability concepts
3	Sustainable development, Nexus between Technology and Sustainable development
4	Millennium Development Goals (MDGs) and Sustainable Development Goals (SDGs)
5	Clean Development Mechanism (CDM)
6	Air Pollution and its effects
7	Water pollution and its sources
8	Zero waste concept and 3 R concepts in solid waste management
9	Greenhouse effect, Global warming, Climate change, Ozone layer depletion
10	Carbon credits, carbon trading and carbon foot print.
11	Legal provisions for environmental protection.
12	Environmental management standards
13	ISO 14001:2015 frame work and benefits
14	Scope and Goal of Life Cycle Analysis (LCA)
15	Circular economy, Bio-mimicking
16	Environment Impact Assessment (EIA)
17	Industrial Ecology, Industrial Symbiosis
18	Basic concepts of Conventional and non-conventional energy
19	General idea about solar energy, Fuel cells
20	Wind energy, Small hydro plants, bio-fuels
21	Energy derived from oceans and Geothermal energy
22	Basic concept of sustainable habitat
23	Methods for increasing energy efficiency of buildings
24	Green Engineering
25	Sustainable Urbanisation, Sustainable cities, Sustainable transport

ASSIGNMENT QUESTIONS

Assignment-1

Attempt to assess the level of damage to the environment due to your actions that have occurred during your last working day, the last week, the last year. Then estimate the damage you are likely to do in your lifetime if you continue in your present ways.

Use the following examples for the above exercise:

- *Example – Plastic: Plastic bags, plastic ball pens*
- Think about all the articles you use daily that are made from plastic. Plastic plays an important part in our modern lives.
- Make a list of the plastic articles you usually use.
- How can you reduce the amount of plastic you use?
- What effects does plastic have on our environment?
- Where did the plastic come from/ how is it made?
- What happens to it when you throw it away/where does it go?

Assignment-2

Explore the different methods that can be adopted for maintaining a sustainable transport system in your city.

Assignment-3

Assess different type of wastes produced in your house and how they affect your environment. Suggest methods to reduce the production of waste and also to reduce their impacts.

ITL201 DATA STRUCTURES LAB

COURSE INFORMATION SHEET

OGRAMME: INFORMATION TECHNOLOGY	DEGREE: BTECH
URSE: DATA STRUCTURES LAB	SEMESTER: III CREDITS: 2
URSE CODE: ITL 201 GULATION: 2019	COURSE TYPE: CORE
URSE AREA/DOMAIN: PROGRAMMING, DATA RUCTURES	CONTACT HOURS: 3 Lab hours/Week.
RRESPONDING LAB COURSE CODE (IF ANY):	LAB COURSE NAME:NA

SYLLABUS:

1. Develop a C program to implement insertion sort, Selection sort and bubble sort*.
2. Design a program to Implement i) Quick sort ii) Merge sort*.
3. Create Programs for i) Linear Search ii) Binary Search*.
4. Create a menu driven program to implement singly linked list operations with options for insertion, deletion, search and traversal*.
5. Device a menu driven program to implement doubly linked list operations with options for insertion at front, insertion at end, deletion at front, deletion at end and traversal.
6. Apply linked list concept to perform polynomial addition*
7. Simulate first fit, best fit and worst fit memory allocation strategies using linked list*.
8. Develop a program to perform stack operations using i) array ii) linked list*.
9. Perform queue operations using i) array ii) linked list*.
10. Apply stack to perform i) Infix to postfix conversion ii) Postfix evaluation *
11. Develop a program to perform preorder, in-order, post order traversals on binary trees*
12. Construct binary search trees to perform insertion, deletion, search
13. Apply Queue and stack in Breadth First Search and Depth First Search respectively*
14. Device Dijkstra's Algorithm for finding Shortest path
15. Resolve the collisions if any using collision resolution techniques like linear Probing, Random Probing, Double hashing and Quadratic Probing*

(* indicates mandatory experiments.)

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T1	Samanta D., Classic Data Structures, Prentice Hall India, 2/e, 2009.
T2	Ellis horowitz, SartajSahni, Fundamentals of Data structures, GalgotiaBooksource
R1	Horwitz E., S. Sahni and S. Anderson, F
R2	Fundamentals of Data Structures in C, University Press (India), 2008.
R3	Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication,1983.
R4	Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill, 1995.
R5	Peter Brass, Advanced Data Structures, Cambridge University Press, 2008
R6	Lipschuts S., Theory and Problems of Data Structures, Schaum's Series, 1986.

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
	Programming in C	C programming Fundamentals	S2

COURSE OBJECTIVES:

1	Understanding the importance of data structures, abstract data type, and their basic usability in different application,
2	Implementing linear and non-linear data structures using linked lists and arrays.
3	Applying various data structure such as stacks, queues, trees, graphs, etc. to solve various computing problems
4	Identifying suitable data structure and algorithm to solve a real world problem.

COURSE OUTCOMES:

After the completion of the course the student will be able to

CO_No	Course Outcome(CO)	Bloom's Category
CO1	**Compare** various kinds of searching and sorting techniques	level 5: Evaluate
CO2	**Construct** Linear and nonlinear data structures using arrays and linked list	level 6:Create
CO3	**Develop** Programs employing dynamic memory management	level 6:Create
CO4	**Choose** appropriate data structure to solve various computing problems.	level 5: Evaluate

CO5	**Originate** hash tables and collision resolution Techniques	level 6:Create
CO6	**Identify** suitable data structure and algorithm to solve a real world problem.	level 3: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	P0 10	PO 11	PO 12	PS O1	PS O2	PS O3
CO1	3	3	3	2	2	1	-	-	-	-	2	1	1	-	-
CO2	3	3	3	3	3	1	-	-	-	-	2	1	3	-	-
CO3	3	3	3	3	3	1	-	-	-	-	2	1	3	-	-
CO4	3	3	3	3	3	1	-	-	-	-	2	1	2	-	1
CO5	3	3	3	3	3	1	-	-	-	-	2	1	1	-	2
CO6	3	3	3	3	3	1	-	-	-	-	2	1	2	-	2

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	LOW/MEDIUM /HIGH	Justification
CO1-PO1	H	The knowledge of structure and abstract data type can be applied to solve complex problems.
CO1-PO2	H	The Knowledge of stack can be used for many problems
CO1-PO3	H	The knowledge of queue can be used to develop algorithms for scheduling
CO1-PO4	M	The knowledge of searching algorithms can be applied in analysis and interpretation of data
CO1-PSO1	L	These fundamental concepts of CS can be applied to solve complex problems
CO2-PO1	H	Efficient algorithms can be designed based on their time complexity.
CO2-PO2	M	Analysis of algorithms helps to select suitable algorithms and reach valid conclusions.
CO2-PSO1	H	Complexity analysis can be applied in research and other innovative areas.
CO3-PO1	M	The knowledge can be enhanced by implementing the data structure using any programming language
CO3-PSO1	H	The implementation of data structures helps to design solutions to complex engineering problems.
CO4-PO1	H	The knowledge about the various data structures can be applied to solve complex engineering problems.
CO4-PO3	M	The knowledge about various data structures can be applied to design efficient solutions to complex engineering problems
CO4-PSO1	L	The knowledge about various data structures can be applied to design efficient solutions to complex

		engineering problems
CO5-PO1	H	The knowledge of searching and sorting algorithms can be applied to solve complex engineering problems.
CO5-PO2	H	The knowledge of searching and sorting algorithms can be appled to analyze problems and reach conclusions.
CO5-PO3	M	The knowledge of searching and sorting algorithms can be applied to design solutions to complex problems.
CO5-PO4	L	The knowledge of sorting algorithms can be applied in analysis and interpretation of data
CO5-PSO1	L	The knowledge of searching and sorting algorithms can be applied in analysis of problems and design solutions.
CO5-PSO3	M	This fundamental knowledge can be used in research and other areas.
CO6-PO1	H	This helps to design an efficient solution to complex problems.
CO6-PO2	H	This knowledge helps in suitable representations and thereby interpretation of data can be done efficiently
CO6-PSO1	M	The knowledge of data structures help to analyze and design solutions to complex problems.
CO6-PSO3	M	This is a core fundamental concept in CS which can be applied in research area also.

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

SNO	DESCRIPTION
1	SPARSE MATRIX AND ITS OPERATIONS
2	CIRCULAR DOUBLY LINKED LIST

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

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

1	AVL TREES

WEB SOURCE REFERENCES:

1	http://www.cse.iitk.ac.in/users/dsrkg/cs210/applets/sortingII/mergeSort/mergeSort.html
2	www.nptel.iitm.ac.in/video.php?subjectId=106105085
3	www.cse.unt.edu/~rada/CSCE3110/Lectures/Trees.ppt
4	cslibrary.stanford.edu/110/BinaryTrees.pdf
5	cslibrary.stanford.edu/103/LinkedListBasics.pdf
6	www.nptel.iitm.ac.in/video.php?subjectId=106105085
7	www.iitg.ernet.in/cse/?page_id=220

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 **Approved by**
Dr. Preetha K G **(H.O.D)**

Lab Cycle and Schedule

Day 1

1. Write a menu driven C program to implement stack using one dimensional array. Perform the operations on the stack (i) Push (ii) Pop (iii) Is empty (iv) Is full (v) Display.
2. Write a menu driven C program to implement queue using one dimensional array. Perform the operations on the Queue (i) Insertion (ii) Deletion (iii) Is empty (iv) Is full (v) Display.

Day 2

3. Write a C program to perform i) Infix to postfix conversion ii) Postfix evaluation

Day 3

4. Write a menu driven C program to implement the following searching operations:
   (i) Linear Search
   (ii) Binary Search

Day 4

5. Write a menu driven C program to implement a singly linked list and perform the following operations on it:

(i) Insertion
   a. at the beginning
   b. at the end
   c. after a specified node.

(ii) Deletion
   a. at the beginning
   b. at the end
   c. a specified node.

(iii) Display the linked list.

(iv) Search an element in the list

Day 5

6. Write a menu driven C program to perform polynomial addition using linked list.

Day 6

7. Write a menu driven C program to implement stack using Singly Linked list. Perform the operations on the stack (i) Push (ii) Pop (iii) Is empty (iv) Is full (v) Display.

8. Write a menu driven C program to implement Queue using Singly Linked list. Perform the operations on the Queue (i) Insertion (ii) Deletion (iii) Is empty (iv) Is full (v) Display

Day 7

9. Write a menu driven program to perform

(i) Bubble Sort
(ii) Insertion Sort
(iii) Selection Sort
(iv) Quick Sort.
(v) Merge Sort.

Day 8

10. Write a menu driven C program to implement a binary tree using linked list and perform the following operations on it

i) Insert a new node.
ii) Delete a specified node.
iii) Search a specified node.

Day 9

11. Write a menu driven C program to implement a binary tree and perform the following traversals on it

iv) In-order
v) Pre order.
vi) Post-order

Day 10

12. Write a menu driven C program to perform the following operations on a directed graph

(i) DFS
(ii) BFS
(iii) Display (using Adjacency Matrix).

Day 11

13. Write a menu driven C program to implement hash table and the following collision resolution techniques-(i) Linear Probing (ii) Quadratic Probing (iii) Chaining

Day12

14. Simulate first fit, best fit and worst fit memory allocation strategies using linked list.

Day 13

Internal Examination

Continuous Internal Evaluation Pattern:
Attendance : 15 marks
Continuous Assessment : 30 marks
Internal Test (Immediately before the second series test): 30 marks

End Semester Examination Pattern: The following guidelines should be followed regarding award of marks
(a) Preliminary work : 15 Marks
(b) Implementing the work/Conducting the experiment : 10 Marks
(c) Performance, result and inference (usage of equipments and trouble shooting) : 25 Marks
(d) Viva voce : 20 marks

(e) Record : 5 Marks

Lab in Charge

Dr.Preetha K G

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
(iii) Make a copy of the given list
(i) 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) L1 ∩ L2
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

ITL203 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: ITL 203 REGULATION: 2019	COURSE TYPE: CORE
COURSE AREA/DOMAIN: PROGRAMMING	CONTACT HOURS: 3 hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY):	LAB COURSE NAME:

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** 1. Function definitions and access 2. Parameters and arguments 3. 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. **Object Oriented Programming using Python**
   c) Creation of Classes & Instances, method calling
   d) Constructor & Destructor concepts
   e) Implementation of Inheritance

Part B : System Utilities

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	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	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	-	-

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 **Approved by**

Dr. Saritha S

LAB SCHEDULE

Week	Remarks
1	Experiment 1
2	Experiment 1
3	Experiment 2
4	Experiment 2
5	Experiment 3
6	Experiment 3
7	Experiment 4
8	Experiment 5

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.

EXPERIMENT 2

Develop a python program to

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

EXPERIMENT 3

1. Develop a python program to create 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 contains the Title of book and Number of copies of each title. As books are added to 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. Develop a python program to implement the following scenario. Given a file "**data.txt**" with three columns of data separated by spaces. Read it into 3 separate simple sequences.
5. 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.

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 the staging area of the Git 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.

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 three numbers equal to zero

3. Write a Python program to find the single element appears once in a list where every element appears four times except for one.

4. Write a Python program to check a sequence of numbers 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 two 4-digit numbers

8. Write a python program to find the 1000th prime number.

9. Given an unordered list of flights taken by someone, each represented as (origin, destination) pairs, and 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 parenthesis 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 the 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 given string.

8. Write a Python program to print all permutations with 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.