SEMESTER 3

PERIOD: SEPTEMBER 2023-JANUARY 2024

RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Department of Information Technology, Programme: Computer Science & Business Systems

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 Computer Science & Business Systems 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)

Computer Science & Business Systems 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)

Computer Science & Business Systems 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.

1) Assignment Schedule

SI No	Subject Code & Name	Faculty in-charge	Week
1	101009/IT300A_FORMAL LANGUAGE AND AUTOMATA THEORY	Dr. Divya PM	Week 2
2	101009/IT300B COMPUTER ORGANIZATION AND ARCHITECTURE	Ms. Viji Mohan A	Week 2
3	101009/IT300C OBJECT ORIENTED PROGRAMMING	Mr. Mathews Abraham	Week 3
4	101009/IT300D COMPUTATIONAL STATISTICS	Ms. Neethu P R	Week 3
5	101009/IT300E DATABASE MANAGEMENT SYSTEMS	Ms. Shareena Basheer	Week 4
6	101009/EN300F ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE	Dr. Sonia Paul	Week 5
7	101009/IT300A_FORMAL LANGUAGE AND AUTOMATA THEORY	Dr. Divya P M	Week 5
8	101009/IT300B COMPUTER ORGANIZATION AND ARCHITECTURE	Ms. Viji Mohan A	Week 6
9	101009/IT300C OBJECT ORIENTED PROGRAMMING	Mr. Mathews Abraham	Week 7
10	101009/IT300D COMPUTATIONAL STATISTICS	Ms. Neethu PR	Week 8
11	101009/IT300E DATABASE MANAGEMENT SYSTEMS	Ms. Shareena Basheer	Week 8
12	101009/EN300F ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE	Dr. Sonia Paul	Week 9

101009/IT300A
FORMAL LANGUAGE AND AUTOMATA THEORY

2.1 COURSE INFORMATION SHEET

PROGRAMME: **COMPUTER SCIENCE AND BUSINESS SYSTEMS**	DEGREE: **B. TECH**
COURSE: **Formal Language and Automata Theory**	SEMESTER: **Three** CREDITS: 3
COURSE CODE: **101009/IT300A** REGULATION: **2022(AUTONOMOUS)**	COURSE TYPE: CORE
COURSE AREA/DOMAIN: Theoretical Computer Science	CONTACT HOURS: 3(L) / Week
CORRESPONDING LAB COURSE CODE (IF ANY): **No**	LAB COURSE NAME: **NA**

SYLLABUS:

UNIT	DETAILS	HOURS
	Regular Languages & Finite Automata **Introduction:** Alphabet, languages and grammars, productions, and derivation, Chomsky hierarchy of languages. **Regular languages and finite automata:** Regular expressions and languages,	

I	deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, Kleene's theorem, pumping lemma for regular languages, Myhill-Nerode theorem and its uses, minimization of finite automata.	9
II	**Context-free languages and pushdown automata:** Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach Normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.	10
III	**Context Sensitive Grammar and Turing Machine** **Context-sensitive languages:** Context-sensitive grammars (CSG) and languages, Linear bounded automata, and equivalence with CSG. **Turing machines:** The basic model for Turing machines (TM), Turing recognizable (Recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.	10
IV	**Undecidability** Church-Turing thesis, universal Turing machine, the Universal and diagonalization languages, reduction between languages and Rice theorem, undecidable problems about languages.	9

V	**Basic Introduction to Complexity** Introductory ideas on Time complexity of deterministic and nondeterministic Turing machines, P and NP, NP- completeness, Cook's Theorem, other NP - Complete problems.	7
	TOTAL HOURS	45

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T	1. Introduction to Automata Theory, Languages, and Computation John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman.
R	1. Elements of the Theory of Computation, Harry R. Lewis and Christos H. Papadimitriou. Automata and Computability, Dexter C. Kozen. 2. Introduction to the Theory of Computation, Michael Sipser. Introduction to Languages and the Theory of Computation, John Martin. 3. Computers and Intractability: A Guide to the Theory of NP Completeness, M. R. Garey and D. S. Johnson. Elements of the Theory of Computation, Harry R. Lewis and Christos H. Papadimitriou. 4. Automata and Computability, Dexter C. Kozen. Introduction to the Theory of Computation, Michael Sipser. 5. Introduction to Languages and the Theory of Computation, John Martin.

TEXT/REFERENCE BOOKS:

COURSE PRE-REQUISITES: Basic knowledge in Set theory, Functions and Relations.

COURSE OUTCOMES: After the completion of the course the student will be able to:

CO1	Understand Chomsky hierarchy and its applications and design automata for regular languages.
CO2	Design a context free grammar for any given context free language.
CO3	Construct Turing machines and understand their capability.
CO4	Understand the Universal Turing Machine and analyze various undecidable problems.
CO5	Understand and analyze P, NP and NP-complete problems.

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
CO1	3	2	_	_	_	_	_	_	_	_	_	2	_	_	_
CO2	3	1	_	_	_	_	_	_	_	_	_	3	2	_	_
CO3	1	_	3	_	2	_	_	_	_	_	_	1	2	_	_
CO4	3	2	_	_	1	_	_	_	_	_	_	2	1	_	_

CO5	3	3	_	_	_	_	_	_	_	_	_	3	1	_	_

JUSTIFICATIONS FOR CO-PO/PSO MAPPING

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
CO1-PO1	H	The knowledge of mathematical formulation of languages will help the students to formulate mathematical solutions for engineering problems.
CO1-PO2	M	The knowledge of formal language hierarchy and its applications will help the students to apply the same to identify and analyze engineering problems.
CO1-PO12	M	Understanding the computational capabilities of different automata will help to recognize the need for independent and life-long learning in the context of technological change.
CO2-PO1	H	Construction of automaton for regular languages will help the students to formulate abstract models for complex engineering problems.
CO2-PO2	L	Knowledge of regular expressions will help the students to identify and analyze complex problems and formulate solutions using the principles of mathematics.

CO2-PO12	H	Construction of regular expressions and automata for a regular language will help to recognize the need for independent and life-long learning in the context of technological change.
CO2-PSO1	M	Correlating the design of automata to real world applications will aid in formulating engineering problems with similar background and arrive at solutions.
CO3-PO1	L	Designing context free grammars for a given language will help to design and develop solutions to engineering problems obeying specific rules.
CO3-PO3	H	Knowledge of context free grammars and production rules will help the students to apply the same in designing system

		components or processes satisfying specific needs and constraints.
CO3-PO5	M	Designing of production rules for a context free language helps the student to identify and apply appropriate tools and techniques to solve real world computational problems.
CO3-PO12	L	Understanding of context free grammars and pushdown automata will help the student recognize the need for life- long learning compatible

		with technological changes.
CO3-PSO1	M	Evaluative study of context free grammars will help in analysis and synthesis of real time computational problems.
CO4-PO1	H	Knowledge of construction of Turing Machines will help the student to apply the same to solutions of complex
CO4-PO2	M	Understanding the different categories of Turing machines will aid in identifying, analyzing and formulating solutions to complex engineering problems.
CO4-PO5	L	Knowledge of computational capabilities of a Turing machine will help in selecting and applying appropriate tools and techniques to solve complex engineering activities, with an understanding of limitations.
CO4-PO12	M	Study of Turing machine and its capabilities will help the student to recognize the need for life-long study competing the technological advances.
CO4-PSO1	L	Understanding the power of a Turing machine helps the student to identify solutions to real world problems while providing an implementation of the same.
CO5-PO1	H	Complexity study of the basic computational problems will help in design and development of solutions for real time computational problems.

CO5-PO2	H	Study of computational problems and their associated complexity will help to provide valid conclusions of real time computational problems
CO5-PO12	H	Knowledge of undecidable problems will help the student to recognize the need to engage in independent and life-long learning in the broadest context of technological change.
CO5-PSO1	L	Students will be able to identify the complexity associated with the computational problems while providing an implementation of the same

GAPS IN THE SYLLABUS - TO MEET INDUSTRY REQUIREMENTS:

Sl. No.	DESCRIPTION	PROPOSED ACTIONS
1	Characterization of computability functions	NPTEL Video Lectures
2	Decision problems and their relationship to model of computations	NPTEL Video Lectures

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

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

(1) Automata based programming

WEB SOURCE REFERENCES:

1. https://nesoacademy.org/cs/04-theory-of-computation
2. https://www.geeksforgeeks.org/theory-of-computation-automata-tutorials/
3. https://www.javatpoint.com/automata-tutorial
4. https://www.tutorialspoint.com/automata_theory/index.htm
5. https://tutorialspoint.dev/computer-science/theory-of-computation

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 Vidhya PM **Dr. Neeba E. A.**

(Faculty in charge) **(HOD)**

2.2 COURSE PLAN

No	Topic	No. of Lectures
1	**REGULAR LANGUAGES AND FINITE AUTOMATA**	**9 Hours**
1.1	Introduction to formal languages	1 Hour

1.2	Deterministic finite automata	2 Hours
1.3	Non-deterministic finite automata	2 Hours
1.4	Equivalence between DFA and NFA	1 Hour
1.5	Minimization of DFA	1 Hour
1.6	Regular expressions and languages	1 Hour
1.7	Pumping lemma and Myhill-Nerode theorem	1 Hour
2	**CONTEXT FREE LANGUAGES & PUSHDOWN AUTOMATA**	**10 Hours**
2.1	Context-free grammars (CFG) and languages (CFL)	1 Hour
2.2	Chomsky and Greibach Normal forms	1 Hour
2.3	Nondeterministic pushdown automata (PDA) and equivalence with CFG	1 Hour
2.4	Parse trees and ambiguity in CFG	2 Hours
2.5	Pumping lemma for context-free languages	1 Hour
2.6	Deterministic pushdown automata	2 Hours
2.7	Closure properties of CFLs.	2 Hours
3	**CONTEXT SENSITIVE GRAMMAR AND TURING MACHINE**	**10 Hours**
3.1	Context-sensitive grammars (CSG) and languages	1 Hour
3.2	Linear bounded automata	1 Hour

3.3	Linear bounded automata and equivalence with CFG	2 Hours
3.4	The basic model for Turing machines (TM)	2 Hours
3.5	Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages	1 Hour
3.6	Variants of Turing machines, nondeterministic TMs, and equivalence with deterministic TMs	1 Hour

3.7	Unrestricted grammars and equivalence with Turing Machines	1 Hour
3.8	TMs as enumerators	1 Hour
4	**UNDECIDABILITY**	**9 Hours**
4.1	Church-Turing thesis	1 Hour
4.2	Universal Turing machine	1 Hour
4.3	The Universal and diagonalization languages	2 Hours
4.4	Reduction between languages	2 Hours
4.5	Rice Theorem	1 Hour
4.6	Undecidable problems about languages.	2 Hours
5	**BASIC INTRODUCTION TO COMPLEXITY**	**7 Hours**
5.1	Introductory ideas on Time complexity of deterministic and	2 Hours

	nondeterministic Turing machines	
5.2	P and NP, NP- completeness	2 Hours
5.3	Cook's Theorem	1 Hour
5.4	NP -Complete problems.	2 Hours

2.3 ASSIGNMENT QUESTIONS

ASSIGNMENT 1

Date: 2/11/2023 Submission Date: 17/11/2023

1. Explain and formally define DFA, NFA and ϵ-NFA along with their corresponding languages and extended transition functions.
2. Explain Chomsky's Hierarchy of Languages.
3. . Define the following:
   i) Kleene star
   ii) Concatenation
   iii) Reversal
   iv) Union
   v) Language
   vi) Finite Automata
   vii) Transition diagram
   viii) Transition table
   ix) Extended transition function
4. List some of the applications of automata theory.
5. If ∑ = {a, b, c} then write ∑1, ∑2, ∑3, ∑*.
6. Discuss the applications of DFA, NFA and epsilon NFA.
7. Why do we need to convert an NFA to DFA?
8. Explain the closure properties of Regular sets.
9. Prove that regular expression is closed under homomorphism. (Refer closure properties of Regular Languages).

10. Which of the following operations are closed under regular sets? Justify your answer.
    i) Complementation
    ii) Set difference
    iii) String reversal
    iv) Intersection
11. What is a regular expression?
12. Write the properties and rules for regular expressions.
13. Discuss the applications of Regular expressions.
14. State and prove pumping lemma theorem for regular languages.
15. Discuss the applications of the pumping lemma.

ASSIGNMENT 2

Date: 2/12/2023 Submission Date: 20/12/2023

1) Discuss the closure properties of context free languages.
2) Discuss the decision properties of context free languages.
3) Define Context Free Grammar and Context Free Language.
4) List the applications of PDA and CFL.
5) What is a derivation tree? Give an example.

$$G\colon E \rightarrow EE+ \mid EE- \mid EE* \mid EE/ \mid a \mid b$$

6) List conditions required for push down automata to qualify as deterministic push down automata.
7) State and prove the pumping lemma for Context Free Languages.
8) What do you mean by useless symbol in a grammar? Show the elimination of useless symbols with an example.

2.4 TUTORIAL QUESTIONS (Conducted as Graded Homework)

1. Design a DFA for the language accepting strings ending with 01 over $\sum = \{0, 1\}$.

2. Draw a DFA for the language accepting strings ending with 'abb' over ∑= {a, b}.
3. Draw a DFA for the language accepting strings ending with 'abba' over ∑= {a, b}.
4. Draw a DFA for the language accepting strings ending with '0011' over ∑= {0, 1}.
5. Draw a DFA for the language accepting strings starting with 1 and ending with 0, over ∑= {0, 1}.
6. Draw a DFA for the language accepting strings having three consecutive 0s, over ∑={0,1}
7. Design a DFA to accept strings with fifth symbol as 1.
8. Design a DFA to accept binary numbers divisible by 3.
9. Design a DFA to accept binary numbers divisible by 3.
10. Design a DFA to accept the set of all strings starting and ending with different alphabets ∑= {a, b}.
11. Design a DFA that accepts the only input 101 ∑= {0, 1}.
12. Draw an NFA which accepts a string containing "ing" at the end of a string with ∑= {a,. . . ., z}.
13. Design an NFA to accept strings starting and ending with 01, with ∑= {0, 1}.
14. Draw a deterministic and non-deterministic finite automate which accept 00 and 11 at the end of a string containing 0, 1 in it, e.g., 01010100 but not 000111010.
15. Draw a deterministic and non-deterministic finite automate which either starts with 01 or end with 01 of a string containing 0, 1 in it, e.g., 01010100 but not 000111010.

16. Draw a deterministic and non-deterministic finite automate which starts with 01 and ends with 01 of a string containing 0, 1 in it, e.g., 01000101 but not 000111001.

17. Draw a deterministic-finite automaton which recognize a string containing binary representation 0, 1 in the form of multiple 2, e.g., 1010 but not 01101.

18. Draw a deterministic finite automaton which recognize a string containing binary representation 0, 1 in the form of multiple 3, e.g., 1001 but not 1000.

19. Construction of a DFA for the set of string over {a, b} such that length of the string |w| is divisible by 2 i.e, |w| mod 2 = 0.

20. Construction of a DFA for the set of string over {a, b} such that length of the string |w| is not divisible by 2 i.e, |w| mod 2 = 1.

21. Construction of a DFA for the set of string over {a, b} such that length of the string |w| is divisible by 3 i.e, |w| mod 3 = 0.

22. Construction of a minimal DFA accepting a set of strings over {a, b} in which the second symbol from left-hand side is always 'b'.

23. Construction of a minimal DFA accepting set of strings over {a, b} in which every 'a' is followed by a 'b'.

24. Draw a Turing machine to find 1's complement of a binary number.

25. Draw a Turing machine to find 2's complement of a binary number.

26.Design a Turing machine to reverse a string consisting of a's and b's.

101009/IT300B
COMPUTER ORGANIZATION AND ARCHITECTURE

3.1 COURSE INFORMATION SHEET

PROGRAMME: COMPUTER SCIENCE AND BUSINESS SYSTEMS	**DEGREE**: BTECH
COURSE: COMPUTER ORGANIZATION AND ARCHITECTURE	**SEMESTER**: III **CREDITS**: 3
COURSE CODE: 101009/IT300B REGULATION: 2021	**COURSE TYPE**: CORE
COURSE AREA/DOMAIN: COMPUTER HARDWARE	**CONTACT HOURS**: 3+0 (Tutorial) +0 (Lab) =3 hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY): NIL	**LAB COURSE NAME**: NIL

SYLLABUS:

UNIT	DETAILS	HOURS
I	Functional blocks of a computer : CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU: Registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Outlining instruction sets of some common CPUs.	8
II	Data representation : Signed number representation, fixed and floating point representations, character representation. Computer arithmetic: Integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and-add, Booth multiplier, carry save multiplier, etc. Division - restoring and non-restoring techniques, floating point arithmetic, IEEE 754 format.	10
III	Introduction to x86 architecture: CPU control unit design: Hardwired and micro-programmed design approaches, design of	10

	a simple hypothetical CPU. Memory system design: Semiconductor memory technologies, memory organization. Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.	
IV	Peripheral devices and their characteristics Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes – role of interrupts in process state transitions, I/O device interfaces – SCII, USB.	9
V	Pipelining : Basic concepts of pipelining, throughput and speedup, pipeline hazards. Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.	8
TOTAL HOURS		45

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T	M. M. Mano, Computer System Architecture: 3rd ed., Prentice Hall of India, New Delhi, 1993.
T	David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Fifth Edition, Morgan Kaufmann Publishers In; 1997.
T	Carl Hamacher, Computer Organization and Embedded Systems, 6Ed, Mgh, 2012.
R	John P. Hayes, Computer Architecture and Organization, McGraw Hill Education; 3rd edition ,2017.
R	William Stallings, Computer Organization and Architecture: Designing for Performance, 9th edition, Pearson Education, 2013
R	Vincent P. Heuring and Harry F. Jordan, Computer System Design and Architecture, Pearson; 1996
R	KaiHwang, Faye Alye Briggs, Computer architecture and parallel processing, McGraw Hill, 1984.
R	Mano M.M, Digital Logic and Computer Design, PHI, 2004.

COURSE OBJECTIVES:

1	The course is prepared with the view of enabling the learners capable of understanding the fundamental architecture of digital computer. Study of Computer Organization and Architecture is essential to understand the hardware behind the code and its execution at physical level by interacting with existing memory and I/O structure. It helps the learners to understand the fundamentals about computer system design so that they can extend the features of computer organization to detect and solve problems occurring in computer architecture.

COURSE OUTCOMES:

SNO	DESCRIPTION	Blooms' Taxonomy Level
101009/IT300B.1	Recognize the relevance of basic components and demonstrate the control signals required for the execution of a given instruction	Level 2 : Understand
101009/IT300B.2	Use and explain the algorithms and implementation aspects of computer arithmetic on binary and floating point numbers.	Level 3: Apply
101009/IT300B.3	Develop the control logic for a given arithmetic problem and explain the types of memory systems and mapping functions used in memory systems	Level 2 : Understand
101009/IT300B.4	Express the importance of I/O organization in a digital computer	Level 2 : Understand
101009/IT300B.5	Examine the concept of pipelining and parallel processing in a digital computer.	Level 2 : Understand

CO-PO AND CO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
101009/IT300B.1	3	2	3	3	-	-	-	-	-	-	-	3	-	-	1

101009/IT300B.2	3	2	3	-	-	-	-	-	-	3	-	2	2	-	1
101009/IT300B.3	3	3	3	3	-	-	-	-	-	3	-	2	2	-	1
101009/IT300B.4	3	3	3	2	-	-	-	-	-	-	-	2	-	-	1
101009/IT300B.5	3	2	1	1	-	-	-	-	-	-	-	2	2	-	1

JUSTIFICATIONS FOR CO-PO AND CO-PSO MAPPING

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
101009/IT300B.1.PO1	H	By imparting to the students complete and detailed knowledge of the internal structure of a computer through graphical models and real examples
101009/IT300B.1.PO2	M	To ensure the understanding of the inner workings of a computer system for the purpose of developing algorithms to solve problems
101009/IT300B.1.PO3	H	To understand the organization and operations of a system and conduct projects to further cross-examine these aspects
101009/IT300B.1.PO4	H	To imbibe the need for detailed research and investigation of operations and sequencing aspects of control signals to come up with solutions for problems regarding the same
101009/IT300B.1.PO12	H	To instill the knowledge of the internal structure of a computer in the students to enable their work in the field of information technology in the future

101009/IT300B.1.PSO3	L	By thoroughly understanding the internal organization, rudimentary operations of the computer, instruction formats and addressing modes and developing technical competencies in systematic planning, testing, and rendering feasible engineered solutions that serve the society in the long run.
101009/IT300B.2.PO1	H	The students are conveyed the details of operations and sequencing in devices through curriculum-based learning as well as through projects and assignments
101009/IT300B.2.PO2	M	The concepts of control signal sequencing are communicated to the students for their understanding on how to better manipulate the system for different purposes
101009/IT300B.2.PO3	H	Students are encouraged develop plans to design systems using their understanding of the basic operations of a system
101009/IT300B.2.PO10	H	To provide a platform for students to come together and share ideas and help each other to understand the concepts of system operations
101009/IT300B.2.PO12	M	To inculcate learning of the sequencing of control signals and basic system operations within and outside the curriculum for the all-rounded development of the students

101009/IT300B.2.PSO1	M	By implementing basic operations in a computer through well-developed algorithms thus enabling students to better grasp the concepts of control signals
101009/IT300B.2.PSO3	L	To assimilate and apply the theoretical flow of control of various operations and sequencing of control signals and disseminating the same to provide the best IT solutions that fits the need of the target users and society as a whole.
101009/IT300B.3.PO1	H	Imparting the students with the working of registers and the arithmetic logic unit as part of the technical education
101009/IT300B.3.PO2	H	Designing circuits involving registers in supervised learning scenarios to better research their behavior in different situations
101009/IT300B.3.PO3	H	To provide resources and information regarding computer systems and inner divisions like ALUs and registers to help students design and develop new systems with the acquired knowledge
101009/IT300B.3.PO4	H	To equip students with the capability to conduct and organize projects and team activities involving the design of systems with registers and ALU
101009/IT300B.3.PO10	H	To introduce the use of modern tools and support to guide and increase in quality the pipelining process and eliminate any hazards faced

101009/IT300B.3.PO12	M	Aims at helping to achieve basic and complete comprehension of registers and ALU design for use in future
101009/IT300B.3.PSO1	M	Applying knowledge gained to understand and design the arithmetic logic unit as well as the different types of registers prevailing in the CPU
101009/IT300B.3.PSO3	L	To critically analyze and represent the design of registers and the arithmetic logic unit and integrate its procedure to real life applications so as to generate solutions and solve engineering problems without ambiguity.
101009/IT300B.4.PO1	H	To impart the concepts of pipelining and any associated hazards to the students through visual and theoretical representation
101009/IT300B.4.PO2	H	Analysis of different solutions to scheduling-based problems and selection of the best remedy
101009/IT300B.4.PO3	H	Using concepts learned in the classroom to design systems with maximum throughput and develop solutions for obstacles faced during the same
101009/IT300B.4.PO4	M	To ensure the detailed analysis and investigation of all aspects of pipelining such as positive as well as adverse effects
101009/IT300B.4.PO12	M	To stimulate the students' skills in arranging processes for maximum throughput and efficiency in a systematic and orderly manner to be put to use in their lives

101009/IT300B.4.PSO3	L	To assess the concept of pipelining ,its associated hazards and to nurture skills in order to successfully administer its application through developing, implementing and other phases of development with the aid of appropriate engineering technologies so that societal requirements are met.
101009/IT300B.5.PO1	H	To provide the understanding of memory systems and I/O interfacing as part of the curriculum
101009/IT300B.5.PO2	M	Comprehension of the difference between cache and DRAM and complete analysis of problems faced in their design
101009/IT300B.5.PO3	L	To impart knowledge on the working of memory and interfacing to encourage students to design and develop systems
101009/IT300B.5.PO4	L	To ensure thorough comprehension of I/O interfacing and memory through graphical and visual representations to initiate investigation in these fields
101009/IT300B.5.PO12	M	To strive towards the all-rounded and complete perception of the students on matters of cache and DRAM performance and I/O interfacing for use in their lives and careers in the future

101009/IT300B.5.PSO1	M	Comparing and analyzing the performance and effectiveness of memory systems like cache and DRAM by integrating both theoretical and practical knowledge
101009/IT300B.5.PSO3	L	To compare and contrast the performance of memory systems like cache ,DRAM and select appropriate interfacing standards for I/O devices and ignite the technical mind in engineering students to devise IT solutions and work-arounds to cater to the larger good of the society.

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

SNO	DESCRIPTION	PROPOSED ACTIONS
1	Processor Logic Design	NPTEL

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

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

1	Introduction to Microprocessors
2	**Flynn's Taxonomy**

WEB SOURCE REFERENCES:

1	https://www.geeksforgeeks.org/computer-organization-and-architecture-pipelining-set-1-execution-stages-and-throughput/
2	https://www.javatpoint.com/dram-in-computer-organization#:~:text=SRAM%20is%20faster.,storing%20every%20bit%20of%20data.
3	https://nptel.ac.in/courses/106103068

4	https://www.geeksforgeeks.org/memory-hierarchy-design-and-its-characteristics/?ref=lbp
5	https://nptel.ac.in/courses/106106166

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	Micro/Mini/Main 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**

Ms. Viji Mohan A
(Faculty)

Dr. Neeba E A
(HOD)

3.2COURSE PLAN

No	Topic	No. of Lectures
1	**Module 1 : Functional blocks of a computer**	**8 Hours**
1.1	CPU, memory - input-output subsystems - control unit.	1 Hour
1.2	Instruction set architecture of a CPU: Registers - instruction execution cycle.	2 Hours
1.3	RTL interpretation of instructions.	1 Hour
1.4	Addressing modes.	2 Hours
1.5	Instruction set.	1 Hour
1.6	Outlining instruction sets of some common CPUs.	1 Hour
2	**Module 2 : Data representation**	**10 Hours**
2.1	Signed number representation - Fixed and floating point representations.	1 Hour
2.2	Character representation.	1 Hour
2.3	Computer arithmetic: Integer addition and subtraction - ripple carry adder.	1 Hour
2.4	Carry look-ahead adder.	1 Hour
2.5	Multiplication - shift-and-add.	1 Hour
2.6	Booth multiplier.	1 Hour
2.7	Carry save multiplier.	1 Hour

2.8	Division - restoring and non-restoring techniques.	2 Hours
2.9	Floating point arithmetic - IEEE 754 format.	1 Hour
3	**Module 3 : Introduction to x86 architecture**	**10 Hours**
3.1	CPU control unit design**:** Hardwired and micro-programmed design approaches.	2 Hours
3.2	Design of a simple hypothetical CPU.	1 Hour
3.3	Memory system design**:** Semiconductor memory technologies.	2 Hours
3.4	Memory organization.	1 Hour
3.5	Memory interleaving - concept of hierarchical memory organization.	1 Hour
3.6	Cache memory - cache size vs. block size.	1 Hour
3.7	Mapping functions.	1 Hour
3.8	Replacement algorithms - write policies.	1 Hour
4	**Module 4 : Peripheral devices and their characteristics**	**9 Hours**
4.1	Input-output subsystems.	1 Hour
4.2	I/O device interface.	1 Hour
4.3	I/O transfers – program controlled.	1 Hour
4.4	Interrupt driven and DMA.	2 Hours
4.5	Privileged and non-privileged instructions.	1 Hour
4.6	Software interrupts and exceptions.	1 Hour
4.7	Programs and processes - role of interrupts in process state transitions.	1 Hour
4.8	I/O device interfaces - SCII, USB.	1 Hour
5	**Module 5 : Pipelining**	**8 Hours**

5.1	Basic concepts of pipelining.	2 Hours
5.2	throughput and speedup.	1 Hour
5.3	pipeline hazards.	2 Hours
5.4	Parallel Processors: Introduction to parallel processors.	1 Hour
5.5	Concurrent access to memory and cache coherency.	2 Hours

3.3 ASSIGNMENT

ASSIGNMENT NO: 1

Submission Date: 06.11.2023

1. Register R1 and R2 contain values 1800 and 3800 respectively. The word length of the processor is 4 bytes. What is the effective address of the memory operand in each one of the following cases?

a. ADD 100 (R2), R6.
b. LOAD R6, 20 (R1,R2)
c. STORE –(R2), R6
d. SUBTRACT (R2)+, R6

2. Registers R1 and R2 of a computer contain the decimal values 1200 and 4600, we have to find effective address of associated memory operand in each instruction:
   a) Load 20(R1),R5
   b) Move #3000,R5
   c) Store R5,30(R1,R2)
   d) Add -(R2),R5
   e) Subtract (R1)+,R5
3. Consider the program
   Move N,R1
   Clear R0
   Add next number to R0
   Decrement R1
   Branch>0
   Move R0, Sum

What is the target address of branch instruction in relative addressing mode (assume the word length is 32 bits and the memory is byte addressable)

4. Multiply -11 x -8 using Booth Algorithm.

5. Multiply each of the following pairs of signed 2's complement number using Booth's algorithm. In each of the cases assume A is the multiplicand and B is the multiplier.

A=110011 B=101100

6. Divide (1001)2 by (11)2 using restoring division algorithm.

7. Given A=10101, B=00100. Perform A/B using restoring division algorithm

ASSIGNMENT NO: 2

Submission Date: 18.12.2023

1. A block-set-associative cache consists of a total of 64 blocks divided into4block sets. The main memory contains 4096 blocks, each consisting of 128 words.
   a) How many bits are there in a main memory address?
   b) How many bits are there in each of the TAG, SET, and WORD fields?
2. A computer system has a main memory consisting of 1M 16-bit words. It also has a 4K-word cache organized in the block-set-associative manner, with 4 blocks per set and 64 words per block. Calculate the number of bits in each of the Tag, Set, and Word fields.
3. Briefly explain, with diagrams, the different methods for control organization.

101009/IT300C
OBJECT ORIENTED PROGRAMMING

4.1 COURSE INFORMATION SHEET

COURSE CODE	COURSE NAME	L	T	P	CREDIT	YEAR OF INTRODUCTION
101009/IT300C	Object Oriented Programming	2	0	4	4	2021

1. Preamble

- To build an understanding of basic concepts of object oriented programming techniques
- To develop programming skills in C++ programming language
- To implement object oriented techniques using C++ language features.
- To develop software using object oriented programming paradigms

2. Prerequisite

Programming in C

3. Syllabus

Module 1: Procedural programming, An Overview of C

Types Operator and Expressions, Scope and Lifetime, Constants, Pointers, Arrays, and References, Control Flow, Functions and Program Structure, Namespaces, error handling, Input and Output (C-way), Library Functions (string, math, stdlib), Command line arguments, Pre-processor directive

Module 2: difference between C and C++

Single line comments, Local variable declaration within function scope, function declaration, function overloading, stronger type checking, Reference variable, parameter passing – value vs reference, passing pointer by value or reference, Operator new and delete, the typecasting operator,Inline Functions in contrast to macro, default arguments

Module 3: The Fundamentals of Object Oriented Programming

Necessity for OOP, Data Hiding, Data Abstraction, Encapsulation, Procedural Abstraction, Class and Object. More extensions to C in C++ to provide OOP Facilities, Scope of Class and Scope Resolution Operator, Member Function of a Class, private, protected and public Access Specifier, this Keyword, Constructors and Destructors, friend class, error handling (exception

Module 4: Essentials of Object Oriented Programming

Operator overloading, Inheritance – Single and Multiple, Class Hierarchy, Pointers to Objects, Assignment of an Object to another Object, Polymorphism through dynamic binding, Virtual Functions, Overloading, overriding and hiding, Error Handling

Module 5:

Generic Programming**:** Template concept, class template, function template, template specialization, Input and Output: Streams, Files, Library functions, formatted output
Object Oriented Design and Modelling: UML concept, Use case for requirement capturing, Class diagram, Activity diagram and Sequence Diagram for design, Corresponding C++ code from design

4. Text Books

1. Lafore R., *Object Oriented Programming in C++*, Galgotia Publications, 2001.
2. Schildt H., *Teach Yourself C++*, Tata McGraw Hill, 2000.

5. Reference Books

1. Hubbard J. R., *Schaum's Outline of Programming with C++*, McGraw Hill, 2000.
2. Balagurusamy, *Object Oriented Programming with C++*, Tata McGraw Hill, 2008.
3. Stephen D. R., C. Diggins, J. Turkanis and J. Cogswell, *C ++ Cook book*, O'Reilly Media, 2013.
4. Oualline S., *Practical C++ Programming*, 2/e, O'Reilly Media, 2002.
5. Meyers S., Effective C++, Addison Wesley, *Error Control Coding Fundamentals and Applications*, Prentice Hall Inc, 2011

6. Course Outcomes

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

CO 1: Explain Object Oriented Programming concepts.
CO 2: To understand the special features of C++ Programming language
CO 3: To upgrade existing procedure oriented softwares to object oriented based ones
CO 4: Able to Apply & Analyze operator overloading, runtime polymorphism , Generic Programming.
CO 5: Able to Analyze and explore various Stream classes, I/O operations and exception handling.

7. 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	PS O1	PS O2	PS O3
CO 1	2	3	3	-	2	-	-	-	-	-	-	1	1	-	-
CO 2	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-
CO 4	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-
CO 5	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-

8. JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
CO1-PO1	M	Acquire conceptual understanding
CO1-PO2	H	Acquire conceptual understanding
CO1-PO3	H	Acquire conceptual understanding

CO1-PO5	M	Acquire conceptual understanding
CO1-PO12	L	Acquire conceptual understanding
CO1-PSO1	M	Acquire conceptual understanding
CO2-PO1	M	C++ programming methods
CO2-PO2	H	C++ programming methods
CO2-PO3	H	C++ programming methods
CO2-PO5	M	C++ programming methods
CO2-PO12	L	C++ programming methods
CO3-PO1	M	C++ programming methods
CO3-PO2	H	C++ programming methods
CO3-PO3	H	C++ programming methods
CO3-PO5	M	C++ programming methods
CO3-PO12	L	C++ programming methods
CO4-PO1	M	C++ programming methods
CO4-PO2	H	C++ programming methods
CO4-PO3	H	C++ programming methods
CO4-PO5	M	C++ programming methods
CO4-PO12	L	C++ programming methods
CO5-PO1	M	C++ programming methods
CO5-PO2	H	C++ programming methods
CO5-PO3	H	C++ programming methods
CO5-PO5	M	C++ programming methods
CO5-PO12	L	C++ programming methods

9. Assessment Pattern

Learning Objectives	**Continuous Internal Evaluation (CIE)**		**End Semester Examination (ESE out of 100)**
	Internal Examination 1* (50)	**Internal Examination 2* (50)**	
Remember			
Understand	25	25	50
Apply	25	25	50
Analyse			
Evaluate			

Create			

***Internal examination (offline): 50 and Internal examination (online): 25**

9. Mark Distribution

Total	**CIE**				**ESE**
	Attendance	**Internal Examination**	**Assignment/Quiz/ Course Project**	**Total**	
150	10	25	15	50	100

10. 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 a maximum of 2 sub-divisions and carry 14 marks.

WEB SOURCE REFERENCES:

1. https://nptel.ac.in/courses/108102045/
2. https://nptel.ac.in/courses/106105159/

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

Mathews abraham
(Faculty)

Approved by

Dr.Neeba E A
(HOD)

4.2 COURSE PLAN

No	Topic	No. of Lectures
1	**Module 1 : Procedural Programming, An Overview of C**	**7 Hours**
1.1	Types Operator and Expressions.	1 Hour
1.2	Scope and Lifetime - Constants.	1 Hour
1.3	Pointers - Arrays and References.	1 Hour
1.4	Control Flow - Functions and Program Structure.	1 Hour
1.5	Namespaces - error handling - Input and Output (C-way).	1 Hour
1.6	Library Functions (string, math, stdlib).	1 Hour
1.7	Command line arguments - Pre-processor directive.	1 Hour
2	**Module 2 : Difference between C and C++**	**6 Hours**
2.1	Single line comments - Local variable declaration within function scope.	1 Hour
2.2	Function declaration - function overloading - stronger type checking.	1 Hour
2.3	Reference variable - parameter passing - value vs reference.	1 Hour
2.4	Passing pointer by value or reference.	1 Hour
2.	Operator new and delete - the typecasting operator.	1 Hour

5		
2.6	Inline Functions in contrast to macro - default arguments.	1 Hour
3	**Module 3 : The Fundamentals of Object Oriented Programming**	**7 Hours**
3.1	Necessity for OOP - Data Hiding.	1 Hour
3.2	Data Abstraction - Encapsulation - Procedural Abstraction.	1 Hour
3.3	Class and Object. More extensions to C in C++ to provide OOP Facilities.	1 Hour
3.4	Scope of Class and Scope Resolution Operator - Member Function of a Class.	1 Hour
3.5	Private, protected and public Access Specifier - this Keyword.	1 Hour
3.6	Constructors and Destructors - friend class.	1 Hour
3.7	Error handling (exception).	1 Hour
4	**Module 4 : Essentials of Object Oriented Programming**	**6 Hours**
4.1	Operator overloading - Inheritance - Single and Multiple.	1 Hour
4.2	Class Hierarchy - Pointers to Objects.	1 Hour
4.3	Assignment of an Object to another Object.	1 Hour
4.4	Polymorphism through dynamic binding.	1 Hour
4.5	Virtual Functions - Overloading.	1 Hour

4.6	Overriding and hiding - Error Handling.	1 Hour
5	**Module 5 : Generic Programming and Object Oriented Modelling**	**6 Hours**
5.1	Generic Programming**:** Template concept - class template.	1 Hour
5.2	Function template - template specialization - Input and Output: Streams.	1 Hour
5.3	Files - Library functions - formatted output - Object Oriented Design and Modelling.	1 Hour
5.4	UML concept - Use case for requirement capturing - Class diagram.	1 Hour
5.5	Activity diagram and Sequence Diagram for design.	1 Hour
5.6	Corresponding C++ code from design.	1 Hour

4.3 ASSIGNMENT 1

- Distinguish between structures and classes.

- Compare and contrast Procedure oriented programming and Object-Oriented Programming
- Explain the use of the following concepts with suitable examples:

  1. new and delete operators.

  2. Pointers & Pointers to Functions

  3.Pointers to objects

  4.Write a program to read & display the total marks of 'N' students and calculate the class average.(store them under an array).

  5.Write a program to find the largest element from the given matrix.

ASSIGNMENT 2

OPERATOR OVERLOADING :

Implement a class string containing the following functions. Develop C++ program using menu – driven approach.

- Overload+ operator to carry out the concatenation of strings.

- Overload = operator to carry out string copy.

- Overload - operator to carry out sub-string removal.

- Overload <= operator to carry out the comparison of strings.

- Overload ++ operator to change string characters to upper case

- Overload -- operator to change string characters to lower case

- Function to display the length of a string.

2. Write a C++ program to implement following operations using operator overloading using menu driven approach.
1. Addition of two complex numbers
2. subtraction of two complex numbers
3. Multiplication of two complex numbers

3. Write a program to overload ++ operator to
rotate an integer array element to left side. eg. array[5]=[0,1,2,3,4], output of array, if the statement is ++i,where i=2 which is accepted from user, is [2,3,4,0,1]. Write a program with proper exception handling, to ensure the value of i in between 0 to 4.

4.4 TUTORIALS

1. Write an object-oriented C++ program to implement the addition of three complex numbers and subtraction of two complex numbers with separate functions for each operation.
2. Write an object-oriented C++ program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows:

Name	Year of joining	Address
Robert	1994	Aluva
Sam	2000	Trissur
John	1999	Shornnur

3. Write a program to print the area of a rectangle by creating a class named 'Area' having two functions. The first function named 'setDim' takes the length and breadth of the rectangle as parameters and the second function named as 'getArea' returns the area of the rectangle. Length and breadth are passed as parameters in setDim(3,4) function.

4. Write a menu-driven program by creating an 'Employee' class having the following functions and print the final salary.

- 1 - 'getInfo()' which takes the salary, number of hours of work per day of employee as parameters
- 2 - 'AddSal()' which adds Rs.500 to the salary of the employee if it is less than Rs.5000.
- 3 - 'AddWork()' which adds Rs.100 to the salary of the employee if the number of hours of work per day is more than 6 hours.

5. Write an object-oriented C++ program to implement a small-scale medical clinic. Detailed attributed and member functions are given below. Develop a Menu Driven program using Switch – Case.

MENU

1. getPatientInfo
2. putPatientInfo
3. makeAppointment
4. Diagnosis
5. Billing
6. printBill

	Remarks

Name of the Class	**patient**	
Attributes of the Class	Public : patientID PName Private : Address Gender Phone Blood Group doctor_Name app_date DiagnosisInfo MedicineInfo doctorFee medicine_charge Total	
Member Functions	Public : void getPatientInfo() void putPatientInfo() void makeAppointment() void Diagnosis() void Billing() void printBill ()	• getPatientInfo() Get following details • patientID, • Pname, • Address • Gender • Phone • Blood Group • putPatientInfo() Print following details • patientID, • Pname, • Address • Gender • Phone • Blood Group • makeAppointment() Get Information on • doctor_Name • app_date • Diagnosis() Get Information on • DiagnosisInfo

		• MedicineInfo • Billing() Get Information on • doctorFee • medicine_charge • and calculate Total • printBill() Print Bill on Screen

1. Write a program with class my_class with an integer data member a and b, a default constructor, a parameterized constructor, a destructor, and a member function that displays the value of a and b. All constructors and, destructor should display a message (Executing Default Constructor / Executing Parameterised Constructor, Executing Destructor).
2. Create a class Car that has two private data fields: integer liters and double consumption. Its default constructor should set those values to 40 and 0.1, respectively. It should also have a parameterized constructor to set those values. Both constructors should display messages that a car with the given features had been created. Its destructor should display a message that a car with the given features had been destroyed. It should have a double maxDistance() method, that will return the maximum distance that the car can travel on a single tank, calculated as liters/consumption.
3. Create a class VehicleRegistration that has the following public and private data fields:

- Public : RegNo, RegYear
- Private : ChasisNo, VehicleType, Nameofowner
- It should have a parameterized constructor, which passes RegNo and RegYear as arguments. RegNo must be an autoincremented value. In the parameterized constructor, a member function call getData() must be included. It is used for getting ChasisNo, VehicleType, Nameofowner from user.
- The class should have one more member function vehicleTransfer(). This function is invoked for owner change. The user must be prompted for "RegNo" entry from main() and check it from available objects. If “RegNo” present call vehicleTransfer(), it must accept Nameofowner to change the ownership. Otherwise, display vehicle not found.
- Its destructor should display a message that a vehicle with the given features had been destroyed.
- Create two objects and execute.

101009/MA300D

COMPUTATIONAL STATISTICS

5.1 COURSE INFORMATION SHEET

101009/MA300D: COMPUTATIONAL STATISTICS

PROGRAMME: **COMPUTER SCIENCE AND BUSSINESS SYSTEM** DEGREE: **BTECH**

COURSE: **MATHEMATICS** SEMESTER: **III** CREDITS: **4**

COURSE CODE: **101009/MA300D** COURSE TYPE: **CORE**

REGULATION: **2021**

COURSE AREA/DOMAIN: **Creative Development** CONTACT HOURS: **3 hours/Week**.

CORRESPONDING LAB COURSE CODE (IF ANY): LAB COURSE NAME:

SYLLABUS:

UNIT	DETAILS	HOURS
I	**MULTIVARIATE NORMAL DISTRIBUTION** Multivariate Normal Distribution Functions- Conditional Distribution and its relation to regression model-Estimation of parameters	9
II	**DISCRIMINANT ANALYSIS** Statistical background-linear discriminant function analysis-Estimating linear discriminant functions and their properties.	9
III	**PRINCIPAL COMPONENT ANALYSIS** Principal components-Algorithm for conducting principal component analysis-deciding on how many principal components to retain-H-plot.	9
IV	**FACTOR ANALYSIS** Factor analysis model-Extracting common factors-determining number of factors-Transformation of factor analysis solutions-Factor scores.	9

V	**CLUSTERING** Introduction-Types of clustering-Correlations and distances-clustering by partitioning methods-hierarchical clustering-overlapping clustering-K-Means Clustering-Profiling and Interpreting Clusters	9
TOTAL HOURS		45

TEXT/REFERENCE BOOKS:

1. Text Books
1. T.W. Anderson, An Introduction to Multivariate Statistical Analysis, Third Edition, JOHN WILEY & SONS, INC., 2003.
2. J.D. Jobson, Applied Multivariate Data Analysis, Vol I & II, Springer New York, 1991.
3. H. Kres, Statistical Tables for Multivariate Analysis, Springer, 1983.
4. Mark Lutz, Programming Python, Second edition, O'Reilly & Associates, INC., 2001
5. Tim Hall and J-P Stacey, Python 3 for Absolute Beginners, Apress,2009.
6. Magnus Lie Hetland, Beginning Python: From Novice to Professional, Apress,2005.

2. Reference Books

1. D.A. Belsey, E. Kuh and R.E. Welsch, Regression Diagnostics , Identifying Influential Data and Sources of Collinearety,Wiley, 2004.
2. J. Neter, W. Wasserman and M.H. Kutner, Applied Linear Regression Models, 2nd Edition, Richard D Irwin INC, 1989.
3. A.S. Mulaik, The Foundations of Factor Analysis, 2nd edition, CRC Press, 2009.
4. D.C. Montgomery and E.A. Peck, Introduction to Linear Regression Analysis,6th edition, John Wiley & Sons,Inc., 2021.
5. M.R. Anderberg, Cluster Analysis for Applications,Academic Press, 1973.
6. D.F. Morrison, Multivariate Statistical Methods, New York, McGrawHill, 1967.
7. Wes Mc Kinney, Python for Data Analysis, O'Reilly, 2013.

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
	Ideas related to Statistical Distribution. Some familiarity with computing and programming experience using languages.		

COURSE OBJECTIVES:

1	To study about distribution functions.
2	To train on computing principal component analysis.
3	To determine the number of factors transforming solutions.
4	To study different clustering.

COURSE OUTCOMES:

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

CO 1: Study about distribution functions and relation to regression model.
CO 2: Properties of discriminant function.
CO 3: Conducting principal component analysis and deciding on how many principal components to retain-H-plot.
CO 4: Determining number of factors and transforming solutions.
CO 5: Study different types of clustering.

CO-PO AND CO-PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1						2	1					
CO2	3	2	1						2	1					
CO	3	2	1						2	1					

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

JUSTIFICATIONS FOR THE MAPPING

Mapping	LOW/MEDIUM/HIGH	Justification
CO1-PO1	H	Knowledge of multivariate distribution and relation to regression model
C01-PO2	M	Analyzing the problems of conditional distribution and its relation to regression model
CO1-PO3	L	Designing solutions for multivariate problems.
CO1-PO9	M	Multivariate analyzing helps the individuals to function effectively.
CO1-PO10	L	Multivariate analysis are used to write effective reports and design documentation.
CO2-PO1	H	Knowledge of estimating linear discriminant functions
CO2-PO2	M	Analyzing the problems on linear discriminant functions
CO2-PO3	L	Designing solutions of linear discriminant functions.
CO2-PO9	M	Linear discriminant functions helps the individuals to find linear combination features that characterizes events
C02-PO10	L	Linear discriminant functions are used to make effective presentations.
CO3-PO1	H	Knowledge for conducting principal component analysis.
CO3-PO2	M	Analyzing how many principal components to retain.
CO3-PO3	L	Designing of H-plot
CO3-PO9	M	Principal components help the individuals to analyze large datasets.
C03-PO10	L	Principal component analysis are used to write effective reports and design documentation.

CO4-PO1	H	Knowledge of Factor analysis model.
CO4-PO2	M	Analyzing how to extract common factors.
CO4-PO3	L	Designing how to transform factor analysis solutions.
CO4-PO9	M	Factor analysis helps the individuals to reduce a large number of variables into fewer numbers of factors.
C04-PO10	L	Factor analysis are used to write effective reports and design documentation.
CO5-PO1	H	Knowledge about different types of clustering
CO5-PO2	M	Analyzing how to apply partition methods for clustering
CO5-PO3	L	Designing of different types of clustering
CO5-PO9	M	Cluster analysis helps the individuals for statistical data analysis like image analysis.
C05-PO10	L	Cluster analysis are used to make effective presentations.

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

Si NO	DESCRIPTION	PROPOSED ACTIONS	RELEVANCE WITH POs	RELEVANCE WITH PSOs
1	LEVEL OF SIGNIFICANCE, CONFIDENCE, LIMITS	READING	1	
2	CRITICAL REGION, CENTRAL LIMIT THEOREM, CONTINUOS DISTRIBUTIONS	READING	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	MULTIVARIATE ANALYSIS OF OTHER DISTRIBUTION	READING	1	

WEB SOURCE REFERENCES:

1	https://www.youtube.com/watch?v=FgakZw6K1QQ
2	https://www.youtube.com/watch?v=3MnVCX94jJM
3	https://www.youtube.com/watch?v=Se28XHI2_xE

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

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

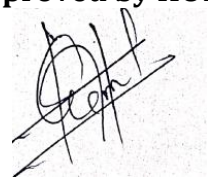
ASSESSMENT METHODOLOGIES-DIRECT

• 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 HOD**



P R NEETHU **DR BINU R**

5.2 COURSE PLAN

No	Topic	No. of Lectures

1	**Multivariate Normal Distribution**	
1.1	Multivariate Normal Distribution Functions	2 Hours
1.2	Conditional Distribution and its relation to regression model	3 Hours
1.3	Estimation of parameters	3 Hours
2	**Discriminant Analysis**	
2.1	Statistical background	1 Hour
2.2	Linear discriminant function analysis	3 Hours
2.3	Estimating linear discriminant functions and their properties	4 Hours
3	**Principal Component Analysis**	
3.1	Principal components	2 Hour
3.2	Algorithm for conducting principal component analysis	2 Hour
3.3	Deciding on how many principal components to retain	2 Hour
3.4	H-plot	3 Hour
4	**Factor Analysis**	
4.1	Factor analysis model	2 Hour
4.2	Extracting common factors	2 Hour
4.3	Determining number of factors	2 Hour
4.4	Transformation of factor analysis solutions	2 Hour
4.5	Factor scores	2 Hour
5	**Clustering**	
5.1	Introduction	1 Hour
5.2	Types of clustering	2 Hour
5.3	Correlations and distances	2 Hour
5.4	Clustering by partitioning methods	1 Hour
5.5	Hierarchical clustering	1 Hour
5.6	Overlapping clustering	1 Hour
5.7	K-Means Clustering	1 Hour

5.8	Profiling and Interpreting Clusters	1 Hour

5.3 Assignment Questions

1. A 100-watt light bulb has an average brightness of 1640 lumens, with a standard deviation of 62 lumens.

a. What is the probability that a 100-watt light bulb will have a brightness more than 1800 lumens?

b. What is the probability that a 100 -watt light bulb will have a brightness less than 1550 lumens?

c. What is the probability that a 100 -watt light bulb will have a brightness between 1600 and 1700 lumens?

2. The average number of acres burned by forest and range fires in a large New Mexico county is 4,300 acres per year, with a standard deviation of 750 acres. The distribution of the number of acres burned is normal.

a. What is the probability that between 2,500 and 4,200 acres will be burned in any given year?

b. What number of burnt acres corresponds to the 38th percentile?

3. The Edwards's Theater chain has studied its movie customers to determine how much money they spend on concessions. The study revealed that the spending distribution is approximately normally distributed with a mean of $4.11 and a standard deviation of $1.37.

a. What percentage of customers will spend less than $3.00 on concessions?

b. What spending amount corresponds to the top 87th percentile?

4. Let $\boldsymbol{XX}$ be distributed as $N_3(\mu, \Sigma N_3(\mu, \Sigma)$, where $\mu^T = (1, -1, 2) \mu^T = (1, -1, 2)$ and $\Sigma = \begin{pmatrix} 4 & 0 & -1 \\ 0 & 5 & 0 \\ -1 & 0 & 2 \end{pmatrix} \Sigma = \begin{pmatrix} 4 & 0 & -1 \\ 0 & 5 & 0 \\ -1 & 0 & 2 \end{pmatrix}$. Which of the following random variables are independent? Explain.

a. X_1X_1 and X_2X_2

b. X_1X_1 and X_3X_3

c. $(X_1, X_3)(X_1, X_3)$ and X_3X_3.

5. Let $\boldsymbol{XX}$ be distributed as $N_3(\mu, \Sigma N_3(\mu, \Sigma)$, where $\mu^T = (1, -1, 2) \mu^T = (1, -1, 2)$ and $\Sigma = \begin{pmatrix} 4 & 0 & -1 \\ 0 & 5 & 0 \\ -1 & 0 & 2 \end{pmatrix} \Sigma = \begin{pmatrix} 4 & 0 & -1 \\ 0 & 5 & 0 \\ -1 & 0 & 2 \end{pmatrix}$. Find the conditional distribution of X_1X_1, given that $X_3 = x_3 X_3 = x_3$.

6. Let $\boldsymbol{XX}$ be distributed as $N_3(\mu, \Sigma N_3(\mu, \Sigma)$ with $\Sigma = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix} \Sigma = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$. Are X_1X_1 and X_2X_2 independent? What about $(X_1, X_2)(X_1, X_2)$ and X_3X_3 ?

7. Consider a bivariate normal population with $\mu_1 = 0\,\mu_1 = 0$, $\mu_2 = 2\,\mu_2 = 2$, $\sigma_{11} = 2, \sigma_{22} = 1 \;\; \sigma_{11} = 2, \sigma_{22} = 1$ and ρ12 = 0.5. Write out the bivariate normal density.

8. The variance covariance matrix of a 3-dimensional random vector $X = (X_1, X_2, X_3)$ is given by $\Sigma = \begin{bmatrix} 25 & -2 & 4 \\ -2 & 4 & 1 \\ 4 & 1 & 9 \end{bmatrix}$. Find the correlation matrix.

9. The weekly distances, D km, a leopard walks in search of food are thought to be Normally distributed with a mean of 62 and a standard deviation of 10. Find the probability that on a given week a leopard will walk less than 80 km.

10. Most graduate schools of business require applicants for admission to take the Graduate Management Admission Council's GMAT examination. Scores on the GMAT are roughly normally distributed with a mean of 527 and a standard deviation of 112. How high must an individual score on the GMAT in order to score in the highest 5%?

11. Why do we need dimensionality reduction? What are its merits and demerits?

12. What will happen when the eigen values are equal in the case of PCA?

13. Can PCA be used to reduce the dimensionality of a highly non-linear dataset? Justify.

14. List out the applications of Principal Component Analysis (PCA).

15. List out the advantages and disadvantages of Principal Component Analysis (PCA).

16. Define the following terms associated with Discriminant Analysis.

i) Canonical Correlation ii) Centroid iii)Classification Matrix iv)Discriminant Scores v)Discriminant function coefficients v) Eigen Values

17. Factory ABC produces very expensive and high quality chip rings that their qualities are measured in terms of curvature and diameter. Result of quality control by experts is given in the following table:

2.9	6.63	Passed

2.53	7.79	Passed
3.57	5.65	Passed
3.16	5.47	Passed
2.58	4.46	Not Passed
2.16	6.22	Not Passed
3.27	3.52	Not Passed

As a consultant to the factory, you get a task to set up criteria for automatic quality control. Solve this problem using Linear Discriminant Analysis.

18. Compute the Linear Discriminant projection for the following two dimensional dataset.

$$X_1 = \{(1, 2), (2, 3), (3.4, 9)\}$$

$$X_2 = \{(2, 1), (3, 2), (4, 3.9)\}$$

19. Consider the two dimensional patterns (0.15,10.5), (0.05,2.5), (0.18,8.5), (0.10,5.5), (0.25,13.5), (0.35,19.5), (0.30, 15.5), (0.22,11.5). Compute the principal component using PCA algorithm

20. a) List out the extensions to Linear Discriminant Analysis(LDA).

b) List out some of the real-world applications of LDA.

101009/IT300E

DATABASE MANAGEMENT SYSTEM

6.1 COURSE INFORMATION SHEET

PROGRAMME: Computer Science & Business Systems	DEGREE: BTECH
COURSE: DATABASE MANAGEMENT SYSTEMS	SEMESTER: S3 CREDITS: 3(3L)
COURSE CODE: 101009/IT300E REGULATION: 2021	COURSE TYPE: CORE
COURSE AREA/DOMAIN: DATABASE MANAGEMENT SYSTEMS	CONTACT HOURS: 3 hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY): 101009/ IT322U	LAB COURSE NAME: Database Management Systems Lab

SYLLABUS:

Module 1: Introduction
Introduction to Database. Hierarchical, Network and Relational Models. Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). Data models: Entity-relationship model, network model, relational and object-oriented data models, integrity constraints, data manipulation operations.
Module 2: Relational Query Languages
Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.
Module 3: Relational Database Design and Query Processing

Relational database design: Domain and data dependency, Armstrong's axioms, Functional Dependencies, Normal forms, Dependency preservation, Lossless design. Query processing and optimization: Evaluation of relational algebra expressions, Query Equivalence, Join strategies, Query optimization algorithms.
Module 4: Storage Strategies and Transaction Processing
Storage strategies: Indices, B-trees, Hashing. Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp-based schedulers, multi-version and optimistic concurrency control schemes, Database recovery.
Module 5: Database Security and Advanced Topics
Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection. Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T	Silberschatz A., H. F. Korth and S. Sudarshan, Database System Concepts, 6/e, McGraw Hill, 2011.
R	1. D Jeffrey Ullman, Principles of Database and Knowledge Base Systems, Volume 1, Computer Science Press, 1988.
R	2. Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education, 2013.
R	3. Serge Abiteboul, Richard Hull, Victor Vianu, Foundations of Databases: The Logical Level, Pearson Education, 1994.
R	4. C.J. Date, A. Kannan, S. Swamynathan, An Introduction to Database Systems, Eighth Edition, Pearson Education, 2006.

R	5. Raghu Ramakrishnan, Database Management Systems, Fourth Edition, McGrawHill College Publications, 2015.
R	6. G.K. Gupta, Database Management Systems, Tata McGraw Hill, 2011.
R	7. Carlos Coronel, Steven Morris, Peter Rob, Database Systems Design, Implementation and Management, Ninth Edition, Thomson Learning, 2009.2.

COURSE PRE-REQUISITES:

Data Structures

Preamble: Database Management Systems course is intended to deliver students the elementary concepts of database management system and equips them to design and implement a database application built over those concepts. This course covers the ER approach to data modeling and the use of query language in SQL. It briefly focuses upon relational database design, query processing and describes the role of transaction management. Database security aspects and advanced topics like web databases, distributed databases, data warehousing and data mining are also covered.

COURSE OUTCOMES:

Students will be able to:

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Describe the fundamental concepts of databases and construct Entity-Relationship	Level 2: Understand
	(ER) models.	
CO 2	Develop queries for relational database in the context of practical applications.	Level 3: Apply
CO 3	Model relational databases following the design principles and perform query optimization.	Level 3: Apply

CO 4	Describe the concepts of storage strategies, control and recovery techniques in transaction processing.	Level 2: Understand
CO 5	Explain the basic concepts of database security and advanced topics like web databases, distributed databases, data warehousing, data mining etc.	Level 2: Understand

CO MAPPING WITH PO, PSO

CO No.	Programme Outcomes (POs)												Programme-Specific Outcomes (PSOs)		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO 1	2	2	3	-	-	-	-	-	-	-	-	-	-	-	-
CO 2	1	2	2	2	3	-	-	-	-	-	-	-	-	**2**	-
CO 3	1	2	3	2	3	-	-	-	-	-	-	2	-	**1**	-
CO 4	1	2	3	-	-	-	-	-	-	-	-	-	-	-	-
CO 5	1	2	2	2	-	-	-	-	-	-	-	-	-	-	-
1010	**1**	**2**	**3**	**2**	**3**							**2**		**2**	

JUSTIFICATION FOR CO-PO-PSO CORRELATION:

JUSTIFICATION FOR CO-PO MAPPING

MAPPING	LEVEL	JUSTIFICATION
101009/IT300E.1-PO1	2	Describing the fundamental concepts of databases and construct Entity-Relationship (ER) models applies the knowledge of engineering.
101009/IT300E.1-PO2	2	Describing the fundamental concepts of databases and construct Entity-Relationship (ER) models identifies
		complex engineering problems reaching substantiated conclusions using engineering sciences.
101009/IT300E.1-PO3	3	Describing the fundamental concepts of databases and construct Entity-Relationship (ER) models designs solutions for complex engineering problems.
101009/IT300E.2-PO1	1	Developing queries for relational database in the context of practical applications applies the knowledge of engineering.
101009/IT300E.2-PO2	2	Developing queries for relational database in the context of practical applications identifies complex engineering problems reaching substantiated conclusions using engineering sciences.
101009/IT300E.2-PO3	2	Developing queries for relational database in the context of practical applications designs solutions for complex engineering problems.

101009/IT300E.2-PO4	2	Developing queries for relational database in the context of practical applications uses research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101009/IT300E.2-PO5	3	Developing queries for relational database in the context of practical applications creates appropriate techniques including prediction and modeling to complex engineering activities with an understanding of the limitations.
101009/IT300E.3-PO1	1	Modeling relational databases following the design principles and perform query optimization applies the knowledge of engineering.
101009/IT300E.3-PO2	2	Modeling relational databases following the design principles and perform query optimization identifies complex engineering problems reaching substantiated conclusions using engineering sciences.
101009/IT300E.3-PO3	3	Modeling relational databases following the design principles and perform query optimization designs solutions for complex engineering problems.
101009/IT300E.3-PO4	2	Modeling relational databases following the design principles and perform query optimization uses research- based knowledge and research methods including design
		of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

101009/IT300E.3-PO5	2	Modeling relational databases following the design principles and perform query optimization creates appropriate techniques including prediction and modeling to complex engineering activities with an understanding of the limitations.
101009/IT300E.3-PO12	3	Modeling relational databases following the design principles and perform query optimization recognizes the need for, and have the preparation and ability to engage in independent and life-long learning.
101009/IT300E.4-PO1	1	Describing the concepts of storage strategies, control and recovery techniques in transaction processing applies the knowledge of engineering.
101009/IT300E.4-PO2	2	Describing the concepts of storage strategies, control and recovery techniques in transaction processing identifies complex engineering problems reaching substantiated conclusions using engineering sciences.
101009/IT300E.4-PO3	3	Describing the concepts of storage strategies, control and recovery techniques in transaction processing designs solutions for complex engineering problems.
101009/IT300E.5-PO1	1	Explaining the basic concepts of database security and advanced topics like web databases, distributed databases, data warehousing and data mining applies the knowledge of engineering.
101009/IT300E.5-PO2	2	Explaining the basic concepts of database security and advanced topics like web databases, distributed databases, data warehousing and data mining identifies complex engineering problems reaching substantiated conclusions using engineering sciences.

101009/IT300E.5-PO3	2	Explaining the basic concepts of database security and advanced topics like web databases, distributed databases, data warehousing and data mining designs solutions for complex engineering problems.
101009/IT300E.5-PO4	2	Explaining the basic concepts of database security and advanced topics like web databases, distributed databases, data warehousing and data mining uses research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and
		synthesis of the information to provide valid conclusions.

JUSTIFICATION FOR CO-PSO MAPPING

MAPPING	LEVEL	JUSTIFICATION
101009/IT300E.2- PSO3	2	Developing queries for relational database in the context of practical applications contribute their engineering skills in engineering domains like database design.
101009/IT300E.3- PSO3	1	Modeling relational databases following the design principles and perform query optimization contribute their engineering skills in engineering domains like database design.

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

SNO	DESCRIPTION	PROPOSED ACTIONS	PO MAPPING
1	Minimal Cover of Functional Dependencies.	Topic beyond syllabus	1, 2, 3, 4

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

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

S No:	DESCRIPTION	PO MAPPING
1	Minimal Cover of Functional Dependencies.	1, 2, 3, 4

DESIGN AND ANALYSIS TOPICS:

Sl. No.	DESCRIPTION	PO MAPPING
1	SQL queries	1, 2, 3, 4

WEB SOURCE REFERENCES:

1.	**https://youtu.be/SkT7jhPAQOE**
2	**https://www.tutorialspoint.com/dbms/index.htm**
3	**https://onlinecourses.nptel.ac.in/noc22_cs91/preview**
4	**https://nptel.ac.in/courses/106106220**
5	**https://archive.nptel.ac.in/noc/courses/noc21/SEM2/noc21-cs58/**
6	**https://onlinecourses.swayam2.ac.in/cec19_cs05/preview**
7	**https://nptel.ac.in/courses/106105175**

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	Micro/Mini/Main 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

6.2 COURSE PLAN:

No	Topic	No. of Lectures
1	**Module 1 : Introduction**	**9 Hours**
1.1	Introduction to Database - Hierarchical, Network and Relational Models.	2 Hours
1.2	Database system architecture: Data Abstraction - Data Independence - Data Definition Language (DDL) - Data Manipulation Language (DML).	3 Hours
1.3	Data models: Entity-relationship model - network model - relational and object-oriented data models - integrity constraints - data manipulation operations.	4 Hours
2	**Module 2 : Relational Query Languages**	**9 Hours**
2.1	Relational Algebra: Relational Algebra Operations - SELECT, PROJECT,	3 Hours

	RENAME - Set Theoretic Operations - JOIN and DIVISION.	
2.2	Tuple and Domain Relational Calculus.	3 Hours
2.3	SQL3 - DDL and DML constructs - Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.	3 Hours
3	**Module 3 : Relational Database Design and Query Processing**	**9 Hours**
3.1	Relational database design: Domain and data dependency - Anomalies in Database Design – Insertion, Deletion and Modification - Armstrong's axioms - Functional Dependencies.	2 Hours
3.2	Normal Forms (NF) - INF, 2NF, 3NF and Boyce - Codd Normal Form - Lossless Join and Dependency Preserving Decompositions.	3 Hours
3.3	Query processing and optimization: Evaluation of relational algebra expressions - Query Equivalence - Join strategies - Query optimization algorithms.	4 Hours
4	**Module 4 : Storage Strategies and Transaction Processing**	**9 Hours**
4.1	Storage strategies: Indices - B-trees - Hashing.	3 Hours
4.2	Transaction processing: Concurrency control - ACID property - Characterizing Schedules: Based on Recoverability and Serializability.	3 Hours
4.3	Locking and timestamp-based schedulers - multi-version and optimistic concurrency control schemes - Database recovery.	3 Hours
5	**Module 5 : Database Security and Advanced Topics**	**9 Hours**
5.1	Database Security: Authentication - Authorization and access control - DAC, MAC and RBAC models - Intrusion detection - SQL injection.	4 Hours
5.2	Advanced topics: Object oriented and object relational databases - Logical databases - Web databases - Distributed databases.	3 Hours
5.3	Data warehousing and data mining.	2 Hours

6.3 ASSIGNMENT 1:

Consider the following database with primary keys underlined (9)

Suppliers (*sid, sname, address*)
Parts (*pid, pname, color*)
Catalog (*sid, pid, cost*)

sid is the key for Suppliers, *pid* is the key for Parts, and *sid* and *pid* together form the key for Catalog. The Catalog relation lists the prices charged for parts by Suppliers.

Write SQL query for

i) Find then names of suppliers who supply some red part
ii) Find the *sids* of suppliers who supply some red or green part
iii) Find the *sids* of suppliers who supply some red part and some green part.

Create the above tables with foreign key constraint, insert values and execute the queries. Illustrate DELETE and UPDATE with example in the above tables.

Consider the following relations: (9)

FACULTY(FNO, NAME, GENDER, AGE, SALARY, DNUM)
DEPARTMENT(DNO, DNAME, DPHONE)
COURSE(CNO, CNAME, CREDITS, ODNO)
TEACHING(FNO, CNO, SEMESTER)

DNUM is a foreign key that identifies the department to which a faculty belongs. ODNO is a foreign key identifying the department that offers a course.Write SQL expressions for the following queries:
(a) Names and department names of faculty members. (b) Names of faculty members not offering any course. (c) Names of departments offering more than three courses, in alphabetic order.

Create the above tables with foreign key constraint, insert values and execute the queries. Illustrate DELETE and UPDATE with example in the above tables.

1.

What are the basic data types available for attributes in SQL? (3)

List the aggregate functions in SQL. (3)

STUDENT(ROLLNO, NAME, AGE, GENDER, ADDRESS, ADVISOR)
COURSE(COURSEID, CNAME, CREDITS)
PROFESSOR(PROFID,PNAME, PHONE)
ENROLLMENT(ROLLNO, COURSEID, GRADE)

Primary keys are underlined. ADVISOR is a foreign key referring to PROFESSOR table. ROLLNO and COURSEID in ENROLLMENT are also foreign keys referring to THE primary keys with the same name.

(i) Names of female students
(ii) Names of male students along with adviser name
(iii) Roll Number and name of students who have not enrolled for any course.

Create the above tables with foreign key constraint, insert values and execute the queries.
Illustrate DELETE and UPDATE with example in the above tables.

2.

Consider the following relations for bank database (Primary keys are underlined):

Customer (customer-name, customer-street, customer-city)
Branch (branch-name, branch-city, assets)
Account (account-number, branch-name, balance)
Depositor (customer-name, account-number)
Loan (loan-number, branch-name, amount)

Answer the following in SQL:

i) Create tables with primary keys and foreign keys (5)

ii) Create an assertion for the sum of all loan amounts for each branch must be less than the sum of all account balances at the branch. (4)

Create the above tables with foreign key constraint, insert values and execute the queries.
Illustrate DELETE and UPDATE with example in the above tables.

3.

For the relation schema below, give an expression in SQL for each of the queries that follows: (9)

employee(employee-name, street, city)
works(employee-name, company-name, salary)
company(company-name, city)
manages(employee-name, manager-name)

a) Find the names, street address, and cities of residence for all employees who work for the Company 'RIL Inc.' and earn more than $10,000.
b) Find the names of all employees who live in the same cities as thecompanies for which they work.
c) Find the names of all employees who do not work for 'KYS Inc.'. Assume that all people work for exactly one company.
d) Find the names of all employees who earn more than every employeeof 'SB Corporation'. Assume that all people work for at most one company.

Create the above tables with foreign key constraint, insert values and execute the queries. Illustrate DELETE and UPDATE with example in the above tables.

4.

In the following tables ADVISOR and TAUGHTBYare foreign keyd referring to the table PROFESSOR. ROLLNO and COURSEID in ENROLLMENT refer to tables with primary keys of the same name.
STUDENT(ROLLNO, NAME, AGE, GENDER, ADDRESS, ADVISOR)
COURSE(COURSEID, CNAME, TAUGHTBY, CREDITS)
PROFESSOR(PROFID,PNAME, PHONE)
ENROLLMENT(ROLLNO, COURSEID, GRADE)

Write SQL expressions for the following queries:
(i) Names of courses taught by 'Prof. Raju'.
(ii) Names of students who have *not* enrolled for any course taught by 'Prof. Ganapathy'.
(iii) For each course, name of the course and number of students enrolled for the course.

Create the above tables with foreign key constraint, insert values and execute the queries. Illustrate DELETE and UPDATE with example in the above tables.

5.

Consider two tables STUDENT(ROLLNO,NAME,CLASS) and (3) ENROLLMENT(ROLLNO,COURSENAME) where ROLLNO in ENROLLMENT is a foreign key referring to STUDENT. It is required that every time a STUDENT tuple is deleted, all the ENROLLMENT tuples referring

to the deleted STUDENT tuple are also deleted. Write SQL statements to specify this foreign key requirement.

a. Find the list of student who has enrolled some courses.
b. Find the list of students who has not enrolled for at least one course

Create the above tables with foreign key constraint, insert values and execute the queries. Illustrate DELETE and UPDATE with example in the above tables.

6.

Consider the following relations:
FACULTY(FNO, NAME, GENDER, AGE, SALARY, DNUM)
DEPARTMENT(DNO, DNAME, DPHONE)
COURSE(CNO, CNAME, CREDITS, ODNO)
TEACHING(FNO, CNO, SEMESTER)

DNUM is a foreign key that identifies the department to which a faculty belongs. ODNO is a foreign key identifying the department that offers a course.

Write SQL expressions for the following queries:

a) Course numbers and names of 3-credit courses offered by 'CS' department.
b) Names of faculty members teaching *maximum*3 courses.
c) Names of departments along with number of courses offered by each of them, in the increasing order of number of courses; exclude departments which do not offer any course.

7.

For the relations listed below, write SQL statements for the updates that follow. (Assume suitable domains for attributes.) (5)

ALBUMS(ALBUM-ID, ALBUM-NAME, PRODUCED-BY, YEAR)

SONGS(SONG-ID, SONG-START, DURATION, ALBUM-ID)

Update the year of the album with name 'SUHANA RATH' to 2018.

Delete the album 'YADON KI BAARISH' along with all the songs in it.

a. Find the list of producers who has produces more than 3 songs which is having duration greater than 3 minutes
b. List the songs with 5 minutes duration in year 2018

ASSIGNMENT II:

1. Goals of data mining and knowledge discovery?
   What are the types of knowledge produced from data mining? Level2.
2. Explain the step-by-step procedure of building a data warehouse.
   Level 2
3. Write about Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design.
   Level 2
4. Write about the different functionalities of a data warehouse.
   Level 2
5. What is a web database? Explain different characteristics of web a database.
   Level 2

101009/EC322G

COMPUTER ORGANISATION & ARCHITECTURE LAB

7.1 COURSE INFORMATION SHEET

PROGRAMME: COMPUTER SCIENCE AND BUSINESS SYSTEMS	DEGREE: B. TECH (Autonomous)
COURSE: **COMPUTER ORGANIZATION & ARCHITECTURE LAB** (offered by DEC)	SEMESTER: 3 CREDITS: 2
COURSE CODE: **101009/EC322G** REGULATION: 2021	COURSE TYPE: CORE
COURSE AREA/DOMAIN: ELECTRONIC CIRCUITS	CONTACT HOURS: 4 hrs.
CORRESPONDING THEORY COURSE CODE (IF ANY): 101009/IT300B	THEORY COURSE NAME: COMPUTER ORGANIZATION & ARCHITECTURE

SYLLABUS:

List of Experiments:

PART A

Circuits on breadboard or simulators

i. Implementation of Combinational Digital/Boolean Circuits: Adder, Subtractor, Multiplication Module, Division Module, Multiplexer, Demultiplexer, Encoder, Decoder.

ii. Implementation of Sequential Circuits: Counters, Linear Feedback Shift Registers (LFSR)

PART B

Machine language programming on x86 or higher version kits or simulators:

i. Add/subtract/multiplication/division/GCD/LCM
ii. Accessing some specific memory locations/ports
iii. Counting odd and even integers from a series of memory locations
iv. Printing values of selected registers
v. Handing interrupts

Text Books

1. M. M. Mano "*Computer System Architecture*", 3rd ed., Prentice Hall of India, New Delhi, 1993.
2. David A. Patterson and John L. Hennessy "*Computer Organization and Design*": The Hardware/Software Interface.
3. Carl Hamacher,"*Computer Organization and Embedded Systems*".
4. Mano M.M.,CilettiM.D.,"*Digital Design*", 4th Edition Pearson India,.2006
5. D.V.Hall, *Digital Circuits and Systems*, Tata McGraw Hill,1989.
6. Barry B Brey, *The Intel Microprocessors: Architecture, Programming and Interfacing*, Prentice Hall of India, NewDelhi, 8th Edition.
7. Kenneth Ayala, *The 8086 Microprocessor: Programming & Interfacing The PC*, Cengage Learning, 1st edition, 2007.

Reference Books

1. John P. Hayes," Computer *Architecture and Organization"*.
2. William Stallings," Computer *Organization and Architecture: Designing for Performance"*.
3. Vincent P. Heuring and Harry F. Jordan," Computer *System Design and Architecture".*
4. A. Anand Kumar, *Fundamentals of Digital Circuits*, 2nd Edition, Prentice Hall,2016.
5. Fletcher, William I., *An Engineering Approach to Digital Design*,1st Edition, Prentice Hall India,1980.

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME
100001/EC200D	Principles of Electronics Engineering
101009/EC222U	Principles of Electronics Engineering Lab

COURSE OBJECTIVES:

1	Familiarize students with the Digital Logic Design through the implementation of Logic Circuits using ICs of basic logic gates.
2	Familiarize students with the Machine language programming on x86 or higher version kits or simulators.

Course Outcomes

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

Sl. No	Descriptions	Blooms Taxonomy level
CO1	**To design and develop** combinational circuits such as Adder, Subtractor, Multiplexer, Demultiplexer, Encoder, Decoder	Understand & Apply (Level 2, 3)
CO2	**To design and develop** sequential circuits such as counters and LFSR.	Understand & Apply (Level 2, 3)
CO3	**Develop** the control logic for a given arithmetic problem and **explain** the types of memory systems and mapping functions used in memory systems.	Understand & Apply (Level 2, 3)

Mapping of Course Outcomes with Program Outcomes

CO No.	Programme Outcomes (POs)												Programme-specific Outcomes (PSOs)		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
1	3			3	-	-	-	-	-	-	-	2	2		
2	3	3	3	3	-	-	-	-	-	3		2	2		
3	3	3	3	3	-	2	-	1	-	3		2	2		
101009 /EC322 G	3	3	3	3		2		1		3		2	2		

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

	PO1	**PO2**	**PO3**	**PO4**	**PO6**	**PO8**	**PO10**	**PO12**	**PSO1**

CO 1	Basics for real-world digital circuit analysis			Investigations of complex problems and research based analysis				Lays foundation for more advanced topics	Principles & techniques learnt can be extended to many future courses like electronic circuits, digital design etc.
CO 2	Applying digital logic to a specific engineering problem	Helps to do complex engineering problems	Development of solutions	Interpret the results and analyse the data			Learn to document the design, communicate the same as reports	Can be used this knowledge in final/interim projects	Principles & techniques learnt can be extended to computer systems related courses
CO 3	**Develop** the control logic for a given arithmetic problem.	Helps to do complex engineering problems	Development of control logic	Interpret the results and analyse the data	Societal applications can be designed	Ethical principle are to be followed	Learn to document the design, communicate the same as reports	Leads to more complex topics digital system design	Principles & techniques learnt can be extended to many

									future courses like VLSI design

Mark Distribution

Total Marks	Continuous Internal Evaluation (CIE)	End Semester Examination (ESE)	ESE duration
150	**75**	**75**	**3 hours**

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 : 25 Marks
d. Viva voce : 20 marks
e. Record : 5 Marks

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

Sl. No	DESCRIPTION	PROPOSED ACTIONS	PO Mapping
1	Self-starting counters & code converter	Reading Assignment (ungraded)	1, 2, 3, 4, 5, 12

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

WEB SOURCE REFERENCES:

1	Self staring counter: courses.cs.washington.edu
2	NPTEL: Electronics - Digital Circuits and Systems, IIT Madras
3	Coursera: Digital Systems: From Logic Gates to Processors

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

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

ASSESSMENT METHODOLOGIES-DIRECT

• ASSIGNMENTS	• ~~STUD. SEMINARS~~	• ~~STUD. VIVA~~	• UNIV. EXAMINATION
• ~~STUD. LAB PRACTICES~~	• TESTS/MODEL EXAMS	• ~~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~~

7.2 COURSE PLAN

1. FAMILIARIZATION OF LOGIC GATES
2. REALIZATION OF LOGIC GATES USING UNIVERSAL GATE (NAND GATES)
3. DESIGN AND REALIZATION OF HALF /FULL ADDER USING BASIC GATES AND UNIVERSAL GATES.
4. DESIGN AND REALIZATION OF SUBTRACTOR USING BASIC GATES AND UNIVERSAL GATES.
5. DESIGN AND REALIZATION OF MULTIPLEXER AND DEMULTIPLEXER USING LOGIC GATES AND ICS.
6. DESIGN AND REALIZATION OF ENCODER AND DECODER USING LOGIC GATES AND ICS.
7. REALIZATION OF FLIP FLOPS: S-R, D, T, JK AND MASTER SLAVE JK FF USING NAND GATES.
8. REALIZATION OF ASYNCHRONOUS COUNTER: 3 BIT UP/DOWN COUNTER.
9. REALIZATION OF SYNCHRONOUS COUNTER: 3 BIT UP/DOWN COUNTER.
10. REALIZATION OF SHIFT REGISTERS USING D FLIP FLOP.
11. REALIZATION OF LINEAR FEEDBACK SHIFT REGISTERS.

12. ADD/SUBTRACT/MULTIPLICATION/DIVISION/GCD/LCM.
13. ACCESSING SOME SPECIFIC MEMORY LOACATIONS/PORTS.
14. COUNTING ODD AND EVEN INTEGERS FROM A SERIES OF MEMORY LOCATIONS.
15. PRINTING VALUES OF SELECTED REGISTERS.
16. HANDING INTERRUPTS.

7.3 LAB CYCLE

PART-A

1. FAMILIARIZATION OF LOGIC GATES
2. REALIZATION OF LOGIC GATES USING UNIVERSAL GATE (NAND GATES)
3. DESIGN AND REALIZATION OF HALF /FULL ADDER USING BASIC GATES AND UNIVERSAL GATES.
4. DESIGN AND REALIZATION OF SUBTRACTOR USING BASIC GATES AND UNIVERSAL GATES.
5. DESIGN AND REALIZATION OF MULTIPLEXER AND DEMULTIPLEXER USING LOGIC GATES AND ICS.
6. DESIGN AND REALIZATION OF ENCODER AND DECODER USING LOGIC GATES AND ICS.
7. REALIZATION OF FLIP FLOPS: S-R, D, T, JK AND MASTER SLAVE JK FF USING NAND GATES.
8. REALIZATION OF ASYNCHRONOUS COUNTER: 3 BIT UP/DOWN COUNTER.
9. REALIZATION OF SYNCHRONOUS COUNTER: 3 BIT UP/DOWN COUNTER.
10. REALIZATION OF SHIFT REGISTERS USING D FLIP FLOP.
11. REALIZATION OF LINEAR FEEDBACK SHIFT REGISTERS.

PART-B

MACHINE LANGUAGE PROGRAMMING ON X86 OR HIGHER VERSION KIT OR SIMULATOR.

1. ADD/SUBTRACT/MULTIPLICATION/DIVISION/GCD/LCM.
2. ACCESSING SOME SPECIFIC MEMORY LOACATIONS/PORTS.
3. COUNTING ODD AND EVEN INTEGERS FROM A SERIES OF MEMORY LOCATIONS.
4. PRINTING VALUES OF SELECTED REGISTERS.
5. HANDING INTERRUPTS.

7.4 OPEN ENDED QUESTIONS

1. Implement the following expression $f(A,B,C,D)=\sum m(0,1,3,4,5,6,7,13,15)$ with minimum number of universal gates.

2. Implement a digital circuit to add 2 bit numbers using NAND gates.
3. Implement 3 bit Binary to Gray & Gray to Binary code converter using mode control.
4. Set up a digital circuit to light 4 LEDs in a row. The lighting pattern is first & second, second & third, fourth & first with an interval of 1sec. This must repeat continuously. Use min: hardware.
5. Set up a 3-bit ring/Johnson counter using mode control (Use min: hardware)
6. Set up a self-starting ring counter.
7. Set up a sequence counter to generate sequences 3, 4, 2, 2, 5....
8. Set up a circuit to display 00 to 99.
9. Set up a circuit to display 00 to 25.
10. Set up a circuit to display 00, 01, 02, 03, 04, 10, 11, 12, 13, 14, 20...94.
11. Implement the following expression f(A,B,C,D)=∑m(0,1,3,4,5,6,7,13,15) using single IC 74151.
12. Implement the following multi output combinational circuit using IC 74138

a. F1(A, B, C, D) =∑m (11,12,13)
b. F1(A, B, C, D) =∑m (1,2,3)
c. F1(A, B, C, D) =∑m (4,5,11)
d. F1(A, B, C, D) =∑m(2,14,5)

13. Set up a circuit that divide clock frequency by 9.
14. Set up a circuit generate sequences 9,4,10,13,6,11,5,2 (Use min: hardware).
15. Set up a circuit to display 00, 01, 10,11,20,21,30,31....90,91.
16. Develop an 8: 1 multiplexer using gates, simulate, test and make it into a subcircuit.
17. Design the circuit of a three bit carry look ahead adder.
18. Design MOD 12 asynchronous counter using T flip-flop.

7.5 ADVANCED QUESTIONS

1. Design a digital system to keep score for a bowling game. The score should be displayed on a 10-bit register in BCD form rather than in binary. Also write a testbench to test your design.
2. Design a digital system to implement a digital clock and write its testbench.
3. Design a 4-way traffic light controller using Verilog HDL. Also write a testbench to test your design.
4. Design a vending machine with four items and provision to return balance amount using Verilog HDL. Also write a testbench to test your design.
5. Design and 8-bit ALU using Verilog HDL and write a testbench for your design.

101009/IT322S
OBJECT ORIENTED PROGRAMMING LAB

8.1 COURSE INFORMATION SHEET

COURSE CODE	COURSE NAME	L	T	P	CREDIT	YEAR OF INTRODUCTION
101009/IT322S	Object Oriented Programming Lab	2	0	4	4	2021

1. Preamble

- To build an understanding of basic concepts of object oriented programming techniques
- To develop programming skills in C++ programming language
- To implement object oriented techniques using C++ language features.
- To develop software using object oriented programming paradigms

2. Prerequisite

Programming in C

3. Syllabus

1. Parameter passing: passing parameter by value vs by reference, passing array as constant pointer
2. Function overloading: writing string operations like strcat and strncat, strcpy and strncpy as overloaded functions.
3. Dynamically allocating space for a pointer depending on input and doing this repeatedly, depending on different inputs and finally de-allocating the pointer.
4. Define class complex with all possible operations: constructor, destructor, copy constructor, assignment operator with the data members stored as pointer to integers.
5. Define class vector of integers with all possible operations like constructor, destructor, copy constructor and assignment operators
6. Define class matrix of integers with all possible operations like constructor, destructor, copy constructor and assignment operators
7. Define class matrix of integers using vector, with all possible operations like constructor, destructor, copy constructor and assignment operators

8. Define class stack, queue, linked-list, array, set using some data-type (int) with data members kept as private and functions kept in both protected and public sections.
9. Define class complex with all possible operators: constructor, destructor, copy constructor, assignment operator and operators >, <, >=, <=, ==, ++ (pre and post), +, +=, ( ), with the data members stored as pointer to integers.
10. Define class vector of integers with all possible operations like constructor, destructor, copy constructor and assignment operators>, <, >=, <=, ==, ++ (pre and post), +, +=, ( )
11. Define class matrix of integers with all possible operations like constructor, destructor, copy constructor and assignment operators>, <, >=, <=, ==, ++ (pre and post), +, +=, ( ).
12. Define class matrix of integers using vector, with all possible operations like constructor, destructor, copy constructor and assignment operators>, <, >=, <=, ==, ++ (pre and post), +, +=, ( ).
13. Define stack and queue inherited from array class, with standard functions and operators
14. Define a class called 'array' with data type passed as template type with constructor, destructor, copy constructor and assignment operators and index operator.
15. Define template functions for compare and use it in the algorithms like bubble sort, insertion sort, merge sort.
16. Formatted input-output examples
17. Input manipulators
18. Overriding operators <<, >>
19. Define class model for complex number, student class, book class and show it using UML diagram as well as concrete class.
20. Show behavioural modelling through sequence diagram and activity diagram for workflow in a typical log-in, log-out situation.

4. Text Books

1. Bjarne Stroustrup,*The C++ Programming Language,* Fourth Edition, Addison Wesley, 2013.
2. Debasish Jana, *C++ and Object-Oriented Programming Paradigm*, Prentice Hall of India, 2003.

5. Reference Books

1. Bjarne Stroustrup, *Programming – Principles and Practice Using C++* , Second Edition, Addison Wesley, 2014.
2. Bjarne Stroustrup, *The Design and Evolution of C++*, Addison-Wesley, 1994.
3. Stephen D. R., C. Diggins, J. Turkanis and J. Cogswell, *C ++ Cook book*, O'Reilly Media, 2013.
4. Oualline S., *Practical C++ Programming*, 2/e, O'Reilly Media, 2002.
5. Meyers S., *Effective C++*, Addison Wesley, 2011.

6. Course Outcomes

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

CO 1: Explain Object Oriented Programming concepts.
CO 2: To understand the special features of C++ Programming language
CO 3: To upgrade existing procedure oriented softwares to object oriented based ones
CO 4: Able to Apply & Analyze operator overloading, runtime polymorphism , Generic Programming.
CO 5: Able to Analyze and explore various Stream classes, I/O operations and exception handling.

7. 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	PS O1	PS O2	PS O3
CO 1	2	3	3	-	2	-	-	-	-	-	-	1	1	-	-
CO 2	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-
CO 4	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-
CO	2	3	3	-	2	-	-	-	-	-	-	-	-	-	-

5															

8. JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
CO1-PO1	M	Acquire conceptual understanding
CO1-PO2	H	Acquire conceptual understanding
CO1-PO3	H	Acquire conceptual understanding
CO1-PO5	M	Acquire conceptual understanding
CO1-PO12	L	Acquire conceptual understanding
CO1-PSO1	M	Acquire conceptual understanding
CO2-PO1	M	C++ programming methods
CO2-PO2	H	C++ programming methods
CO2-PO3	H	C++ programming methods
CO2-PO5	M	C++ programming methods
CO2-PO12	L	C++ programming methods
CO3-PO1	M	C++ programming methods
CO3-PO2	H	C++ programming methods
CO3-PO3	H	C++ programming methods
CO3-PO5	M	C++ programming methods
CO3-PO12	L	C++ programming methods
CO4-PO1	M	C++ programming methods
CO4-PO2	H	C++ programming methods
CO4-PO3	H	C++ programming methods
CO4-PO5	M	C++ programming methods
CO4-PO12	L	C++ programming methods
CO5-PO1	M	C++ programming methods
CO5-PO2	H	C++ programming methods
CO5-PO3	H	C++ programming methods
CO5-PO5	M	C++ programming methods
CO5-PO12	L	C++ programming methods

9. Assessment Pattern

Learning Objectives	Continuous Internal Evaluation (CIE)		End Semester Examination (ESE out of 100)
	Internal Examination 1*	Internal Examination 2*	

	(50)	(50)	
Remember			
Understand	25	25	50
Apply	25	25	50
Analyse			
Evaluate			
Create			

***Internal examination (offline): 50 and Internal examination (online): 25**

9. Mark Distribution

Total	CIE				ESE
	Attendance	Internal Examination	Assignment/Quiz/ Course Project	Total	
150	10	25	15	50	100

10. 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 a maximum of 2 sub-divisions and carry 14 marks.

WEB SOURCE REFERENCES:

1. https://nptel.ac.in/courses/108102045/
2. https://nptel.ac.in/courses/106105159/

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**

Mathews Abraham
(Faculty)

Dr.Neeba E A
(HOD)

8.2 COURSE PLAN

Sl.No	Experiment	Schedule
1	Fundamentals of Class and Objects	Week 1
2	Constructor and Destructor	Week 2
3	MULTIPLE OBJECTS	Week 3
4	FRIEND CLASSS	Week 4
5	FRIEND FUNCTION	Week 5
6	FUNCTION OVERLOADING	Week 6
7	INHERITANCE	Week 7
8	INHERITANCE USING ABSTRACT CLASS	Week 8
9	MULTIPLE INHERITANCE	Week 9
10	MULTILEVEL INHERITANCE	Week 10
11	OPERATOR OVERLOADING	Week 11
12	VIRTUAL FUNCTIONS	Week 12
13	MACRO PROJECT	

8.3 Lab Cycle

1. **Fundamentals of Class and Objects** :
   Write a C++ program to implement a Class Student with the following attributes

Name of the Class	Student
Class Members	RNo : Roll Number of the student Sname : Name of the student

	Contact : Contact Number Email : Email Address SemGPA[8] : Array to store Semester wise GPA TotalCGPA : Total CGPA
Member Functions	getData() : Function to get initial values from User putData() : Function to display values in class compute() : Function to Compute CGPA

Implement the program to create one object for the class student.

2. Constructor and Destructor:

Write a C++ program to implement a Class Student with the following attributes

Name of the Class	**Student**	**Remarks**
Class Members	RNo : Roll Number of the student Sname : Name of the student Contact : Contact Number Email : Email Address SemGPA[8] : Array to store Semester wise GPA TotalCGPA : Total CGPA	RNo. must be initialized by Constructor. Public : RNo : Roll Number of the student Sname : Name of the student Private : Contact : Contact Number Email : Email Address SemGPA[8] : Array to store Semester wise GPA TotalCGPA : Total CGPA
Member Functions	getData() : Function to get initial values from User putData() : Function to display values in class compute() : Function to Compute CGPA	getData() : receives data except RNo. compute() is private. Add constructors and destructors where ever applicable.

Implement the program to create multiple objects for the class student using array of objects. RNO should be updated by constructor following an

autoincrement method. Note : Use "static int count=1;" as a global variable and use that value for initialization in the constructor. Use count increment to support autoincrement.

3. MULTIPLE OBJECTS :

Given that an EMPLOYEE class contains the following members:

Name of the Class	**Employee**	**Remarks**
Class Members	ENo : Employee Number Ename : Employee Name Basic : Basic Salary DA : Dearness Allowance IT: Income Tax NetSalary : Net Salary after deductions	ENo must be initialized by Constructor by following autoincrement
Member Functions	getData() : Function to get initial values from User putData() : Function to display values in class computeNetSalary() : Function to Compute Net Salary	getData() : receives data except ENo. getData() and putData() are public. computeNetSalary() is private Net salary = (Basic Salary + DA) - IT DA = 52% Basic IT = 0% if Basic < 10,000 10% if Basic < 20,000 20% if Basic < 30,000 30% if Basic<40,000 Add constructors and destructors where ever applicable.

Write a C++ program to read the data of N employee and compute Net salary of each
employee

4. **FRIEND CLASS** :

Implement a class “Student” with name and roll. No attributes. Derive two derived classes "Internal” and “External” from class Student. “Internal” Class has IA1,IA2,IA3 attributes representing Internal Assessment marks. “External” Class has an ESE attribute representing End Semester marks. Implement a class “Result” by including classes “Internal” and “External” as friend classes. The “Result “Class must have an attribute total, and it must be calculated by a member function, which can access the marks given in the “Internal” and “External” classes.

5. FRIEND FUNCTION :

Assume that a bank maintains two kinds of accounts for customers, one called as savings account and other as current account. Create a class account that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions to achieve the following tasks.

i. Accept deposit from a customer and update the balance.
ii. Display the balance.
iii. Compute and deposit interest.
iv. Permit withdrawal and update balance.

Use Friend function, Write a C++ program to list all accounts of an individual.

6. **FUNCTION OVERLOADING** :

Create a class point with the following instance variable and methods.

Member variables: private int x,y
Constructors : public Point(), Point(int x, int y)
Methods : public void setX(int x), setY(int y), setXY(int x, int y)
public void Sum(x,y);//Sum of x and y
public void Sum(x); //Sum of digits of x/y

7. **INHERITANCE** :

Given that an EMPLOYEE class contains the following members:

Name of the Class	Employee	Remarks
Class Members	Name : Name of the Employee Age : Age of the Employee Phone : Phone Number of the Employee Salary : Salary of the Employee	
Member Functions	printSalary() : Display Salary	

Write a C++ program that creates a class named 'Employee'. Define two classes 'Officer' and 'Manager' which inherit the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same. Use Constructor to read data from the keyboard for both classes.

8. **INHERITANCE USING ABSTRACT CLASS** :

Write a C++ program to create an abstract class named “Shape” that contains an empty method named numberOfSides(). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures.

9. **MULTIPLE INHERITANCE** :

Given the two base classes with the following members.

Name of the Class	Student	Remarks
Class Members	RNo : Roll Number of the student M1 : Mark of subject 1 M2 :Mark of subject 2	
Member Functions	getStudentData() : Function to get initial values from User	

Name of the Class	Sport	Remarks
Class Members	graceMarks : Grace marks for Sports activity	
Member Functions	getSportsData() : Function to get initial values from User	

Create a new class REPORT inherited from student and sports classes.

Name of the Class	REPORT	Remarks
Class Members	Total = M1 + M2 + graceMarks Avg = Total/3	All members from Student and Sports classes
Member Functions	displayReport() : Function to display report	

Write a C++ program to implement the above by using multiple inheritances.

10. **MULTILEVEL INHERITANCE** :
Given the two base classes with the following members.

Name of the Class	Student	Remarks
Class Members	RNo : Roll Number of the student SName : Name of the Student	
Member Functions	getStudentData() : Function to get initial values from User	

Create a new class TEST inherited from the STUDENT class.

Name of the Class	Test	Remarks
Class Members	markSub1 : Mark of Subject1	

	markSub2 : Mark of Subject2 markSub3 : Mark of Subject3	
Member Functions	getMarks() : Function to get initial values from User	

Create a new class RESULT inherited from TEST Class.

Name of the Class	**RESULT**	**Remarks**
Class Members	Total = markSub1+ markSub2+ markSub3	
Member Functions	putResult() : Function to get initial values from User	Function will compute "Total" and print RNO, SName and Total

Write a C++ program to implement the above by using multilevel inheritances.

11. **OPERATOR OVERLOADING** :

Implement a class string containing the following functions:

- Overload + operator to carry out the concatenation of strings.
- Overload = operator to carry out string copy.
- Overload <= operator to carry out the comparison of strings.
- Function to display the length of a string.

12. **VIRTUAL FUNCTIONS** :

Write a program to print the marklist of student using student class representing name and roll. No and two derived classes "subject" and "language". The "subject" and "language" class should be inherited by a "result" class having the functionality to add the marks of both subject and language and print the marklist for a student. Use virtual functions to eliminate the ambiguity.

13. **MACRO PROJECT :** Student group consisting of three members have to analyze, design, develop and demonstrate a given problem. Three members group

shall be formed within the batches, in consultation with the staff in charge of each batch.

8.5 Advanced List of Experiments

a. Write a program to implement Virtual Base Class

b. Write a Program Containing a Possible Exception. Use a Try Block to Throw it and a Catch Block to Handle it Properly.

c. Write a C++ program that uses function templates to find the largest and smallest number in a list of integers and to sort a list of numbers in ascending order.

d. Write a Program to Demonstrate the Catching of All Exceptions.

101009/IT322T
COMPUTATIONAL STATISTICS LAB

9.1 Course Information Sheet

PROGRAMME: Computer Science & Business Systems	DEGREE: BTECH
COURSE: COMPUTATIONAL STATISTICS LAB	SEMESTER: III CREDITS: 1
COURSE CODE: 101009/IT322T **REGULATION:** 2021	COURSE TYPE: CORE
COURSE AREA/DOMAIN: COMPUTATIONAL STATISTICS	CONTACT HOURS: 3 hours/Week.

Syllabus:

UNIT	DETAILS	HOURS

I	**Part A : Python Concepts, Data Structures, Classes** 1. Basic programming experiments on Interpreter, Program Execution, Statements, Expressions. 2. Flow controls 3. Functions 4. Basic programming experiments on Numeric Types, Sequences and Class Definition 5. Constructors 6. Text & Binary Files - Reading and Writing. **Part B : Data Wrangling** 1. Basic programming experiments on Combining and Merging Datasets 2. Perform Reshaping and Pivoting 3. Experiments on Data Transformation 4. Experiments on String Manipulation 5. Experiments on Regular Expressions. **Part C : Data Aggregation, Group Operations, Time series** 1. Experiments on Group by Mechanics 2. Data Aggregation, Groupwise Operations and Transformations 3. Pivot Tables and Cross Tabulations 4. Time Series Basics and Data Ranges 5. Frequencies and Shifting. **Part D : Visualization in Python** 1. Familiarization on Matplotlib packages 2. Perform basic programming experiments on Plotting Graphs, Controlling Graph, Adding Text 3. Experiments on More Graph Types, Getting and setting values and Patches.	

Text/ Reference Books:

T/R	BOOK TITLE/AUTHORS/PUBLICATION

T	1. T.W. Anderson, An Introduction to Multivariate Statistical Analysis, Third edition, Wiley, 2009 2. J.D. Jobson, Applied Multivariate Data Analysis,Springer, Vol I & II, Springer, 1991 3. H. Kris, Statistical Tests for Multivariate Analysis, Springer-Verlag, Heidelberg 4. Mark Lutz, Programming Python, 4th Edition, O'Reilly Media, Inc, 2010 5. Tim Hall and J-P Stacey, Python 3 for Absolute Beginners, Apress Berkeley, CA, 2009
R	1.D.A. Belsey, E. Kuh and R.E. Welsch, Regression Diagnostics, Identifying Influential Data and Sources of Collinearety, Wiley-Interscience, 2004. Neter, W. Wasserman and M.H. Kutner, Applied Linear Regression Models, Graw-Hill Education, 2004. .S. Mulaik, The Foundations of Factor Analysis, 2nd Edition, Chapman and l/CRC, 2009. .C. Montgomery and E.A. Peck, Introduction to Linear Regression Analysis, Fifth Edition, Wiley, 2012. I R. Anderberg, Cluster Analysis for Applications, Elsevier Science Publishing Co Inc, 1973. .F. Morrison, Multivariate Statistical Analysis, Brooks/Cole, 2004. Ves Mc Kinney, Python for Data Analysis, Second Edition, O'Reilly Media, 2017.

COURSE PRE-REQUISITES:

Python Programming Knowledge

COURSE OUTCOMES:

CO No.	Course Outcomes	Bloom's Category
CO 1	Develop readable* Python programs by making use of basic constructs- Interpreter, Program Execution, Statements, Expressions, Flow controls.	Create

CO 2	Design modular Python programs using normal Functions, Numeric Types, Sequences and Class Definition, Constructors, Text & Binary Files.	Create
CO3	Understanding the concepts of Data Wrangling includes Reshaping and Pivoting, Data Transformation, String Manipulation, Regular Expressions.	Understand
CO 4	Understand the visualization of Data Aggregation, Group Operations and Time series.	Understand
CO 5	Understand the implementation of Data Visualization in Python includes Plotting Graphs, Controlling Graph, Adding Text, More Graph Types etc.	Understand
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	1	-	-	-	-	2	2
CO 2	2	3	3	3	3	1	-	-	-	-	1	2
CO 3	2	2	1	2	1	1	-	-	-	-	2	2
CO 4	3	3	3	3	3	1	-	-	-	-	2	3
CO 5	2	1	2	3	3	1	-	-	-	-	2	3

COs	**PSO 1**	**PSO 2**	**PSO 3**

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

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

CO2-PO3	3	Writing modular programs in python to implement various computational tasks will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems.
CO2-PO4	3	Writing modular programs in python to implement various computational tasks will help to conduct investigations of complex problems
CO2-PO5	3	Writing modular programs in python to implement various computational tasks will help to apply IT tools to model complex engineering activities.
CO3-PO1	2	Writing programs in Data Wrangling includes Reshaping and Pivoting, Data Transformation, String Manipulation, Regular Expressions.
CO3-PO2	3	Writing programs in pythons using Data Wrangling includes Reshaping and Pivoting, Data Transformation, String Manipulation, Regular Expressions will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
C03-PO3	3	Writing programs in pythons using Data Wrangling includes Reshaping and Pivoting, Data Transformation, String Manipulation, Regular Expressions will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal

		Problems
CO3-PO4	3	Writing programs in pythons using Data Wrangling includes Reshaping and Pivoting, Data Transformation, String Manipulation, Regular Expressions will help to conduct investigations of complex problems
CO3-PO5	3	Writing programs in pythons using Data Wrangling includes Reshaping and Pivoting, Data Transformation, String Manipulation, Regular Expressions will help to Create, select, and apply appropriate techniques using IT tools

CO4-PO1	2	Experimenting with Data Aggregation, Group Operations and Time series will help to apply the knowledge of engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
CO4-PO2	2	Experimenting with the Data Aggregation, Group Operations and Time series will help to identify, formulate, and analyze complex engineering problems reaching substantiated conclusions using first principles of engineering sciences.
C04-PO3	1	Experimenting with the Data Aggregation, Group Operations and Time series utilities will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO4-PO4	2	Experimenting with the Data Aggregation, Group Operations and Time series will help to conduct investigations of complex Problems
CO4-PO5	1	Experimenting with the Data Aggregation, Group Operations and Time series utilities will help to create, select, and apply appropriate techniques using IT tools
C05-PO3	2	Experimenting with Data Visualization in Python will help to design solutions for complex engineering problems and design system processes that meet the specified needs with appropriate consideration for the societal problems
CO5-PO5	3	Experimenting with Data Visualization in Python will help to create, select, and apply appropriate techniques using IT tools.
CO5-P09	3	Experimenting with Data Visualization in Python will help to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
CO5-P10	2	Experimenting with Data Visualization in Python will help to Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Mapping	**Level**	**Justification**
CO1-	2	Writing programs in python using basic constructs will help to

PSO1		acquire skills to design, analyse and develop algorithms and implement those using Python.
CO2-PSO1	2	Writing modular programs in python programs to implement various computational tasks will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO3-PSO1	2	Writing programs in pythons using Data Wrangling will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO4-PSO1	1	Writing programs in pythons using Data Aggregation will help to acquire skills to design, analyse and develop algorithms and implement those using Python.
CO5-PSO1	1	Writing programs in pythons using Visualisation to implement daily life problems and their solutions acquire skills to design, analyse and develop algorithms and implement those using Python.

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

SL 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	• STUD.	• ADD-ON COURSES

BOARDS	SEMINARS	

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 Vidhya PM **Dr. Neeba E A**
Faculty **HOD**

9.2 Course Plan

No	Topic	No. of hours
1	**Experiment 1 : Python Concepts, Data Structures, Classes**	**9 Hours**

1.1	Basic programming experiments on Interpreter, Program Execution, Statements, Expressions.	2 Hours
1.2	Flow controls.	2 Hours
1.3	Functions.	2 Hours
1.4	Basic programming experiments on Numeric Types, Sequences and Class Definition.	3 Hours
1.5	Constructors.	
1.6	Text & Binary Files - Reading and Writing.	
2	**Experiment 2 : Data Wrangling**	**10 Hours**
2.1	Basic programming experiments on Combining and Merging Datasets	2 Hours
2.2	Perform Reshaping and Pivoting	2 Hours
2.3	Experiments on Data Transformation	2 Hours
2.4	Experiments on String Manipulation	2 Hours
2.5	Experiments on Regular Expressions.	2 Hours
3	**Experiment 3 : Data Aggregation, Group Operations, Time Series**	**10 Hours**
3.1	Experiments on Group By Mechanics.	2 Hours
3.2	Data Aggregation, Groupwise Operations and Transformations.	2 Hours
3.3	Pivot Tables and Cross Tabulations.	2 Hours
3.4	Time Series Basics and Data Ranges.	2 Hours
3.5	Frequencies and Shifting.	2 Hours
4	**Experiment 4 : Visualization in Python**	**5 Hours**
4.1	Familiarization on Matplotlib packages.	1 Hour

4.2	Perform basic programming experiments on Plotting Graphs, Controlling Graph, Adding Text.	2 Hours
4.3	Experiments on More Graph Types, Getting and setting values and Patches.	2 Hours

9.3 Lab Cycle

Lab Cycle 1:Python Concepts, Data Structures, Classes

1. Write a program to accept two numbers from the user and calculate multiplication.
2. Print all prime numbers with in an interval
3. Search an element in a list
4. Find the number of occurrences of a given substring in a string.
5. Implements calculator with functions like add, subtract, multiply, divide, exponent etc.
6. Find factorial of a given number using recursion.
7. Return multiple values from a function
8. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise “Required copies not in stock” is displayed. Design a system using a class called books with suitable member functions, constructors and destructors.
9. Develop a python code to read a text file, copy the contents to another file after removing the blank lines.
10. Find the most frequent words in a text read from a file.

Lab Cycle 2: Data Wrangling

1. Write a Pandas program to add, subtract, multiple and divide two Pandas Series.
2. Write a Pandas program to create a dataframe of Student Details.
3. Create a DataFrame of Student Details containing ID, Name and Branch. Merge the DataFrame with another DataFrame(Fee Details) using common ID.
4. Write a Pandas program to create a car DataFrame with fields Brand, Year, No of Sold. Use Groupby operation to Group the data when year = 2010. Also perform reshaping the DataFrame with stack and unstack operation.
5. Write a Pandas program to create a given DataFrame and set index to it. Use transform() to function to add 10 to each element in the dataframe.

Lab Cycle 3:Data Aggregation, Group Operations, Time series

I. AGGREGATION AND GROUPING

1. Load 'planets' dataset and apply simple aggregation functions to the dataset th at
grouped by the feature 'method' and iterate over the group
   - count() Total number of items
   - first(), last() First and last item
   - mean(), median() Mean and median
   - min(), max() Minimum and maximum
   - std(), var() Standard deviation and variance
   - mad() Mean absolute deviation
   - prod() Product of all items
   - sum() Sum of all items

2. Apply groupby operation on the feature 'method' with aggregation, filt ering,
transform and apply to the planets dataset.
   a) Find the minimum of all attributes
   b) Filter the values whose year value count is greater than 5
   c) Transform the values to a new value that is subtracted with its mean
   d) Apply the same filter function to the dataset using 'apply'

II. PIVOT TABLE

1. Load the titanic dataset and analyze the features, data types, and other details using essential functions.
2. Create a pivot table that provides a data frame of survived female and male passengers in different classes.

a. with single aggfunc

a. with multiple aggfuncs

a. with row and column total values

a. With multiple features like survived and age.

a. With multiple features like sex and deck passed as indices.

III. TIME SERIES

1. List the operations available with datetime, dateutil, and datetime64 functions like parse, strftime, strptime with examples
2. Implement Regular sequences of time series using pd.date_range()
3. Implement Time Series with Indexing by time
4. Implement Time Series Data Structures with datetime, period and timedelta indices
5. Implement Frequencies and Offsets in time series data

Lab Cycle 4:Visualization in Python

1. Draw a Dot Chart for 3 variables for the data.
2. Draw a Pie Chart and Radial Diagram for the data.
3. Draw a Chart Tables for the data.
4. Draw a Histogram overlay for the data.
5. Draw a Box Plots for group for the data.
6. Draw a Scatter Plot for Four Quadrants differentiated by colors for the data.
7. Draw a Scatter Plot for Outliers Highlighted for the data.
8. Draw a Scatter Plot for Areas Highlighted for the data.

9.4 Open Questions

1. Write a Pandas program to add, subtract, multiple and divide two Pandas Series.
2. Write a Pandas program to compare the elements of the two Pandas Series.
3. Write a Pandas program to convert a dictionary to a Pandas series.
4. Write a Pandas program to sort a given Series.
5. Write a Pandas program to create a subset of a given series based on

value and condition.
6. Write a Pandas program to create the mean and standard deviation of the data of a given Series.
7. Write a Pandas program to compute the minimum, median, mean and maximum of a given series.
8. Write a Pandas program to display most frequent value in each series and replace everything else as 'Other' in the series.
9. Write a Pandas program convert the first and last character of each word to upper case in each word of a given series.
10. Write a Pandas program to calculate the number of characters in each word in each series.
11. Program to add two matrices using nested loop
12. Program to transpose a matrix using a nested loop
13. Program to multiply two matrices using nested loops
14. Write a Pandas program to create a car DataFrame with fields Brand, Year, No of Sold. Use Groupby operation to Group the data when year = 2010. Also perform reshaping the DataFrame with stack and unstack operation.
15. Create a DataFrame of Student Details containing ID, Name and Branch. Merge the DataFrame with another DataFrame(Fee Details) using common ID.

9.5 Advanced Questions

Develop a python program to:

1. Develop a python code to read a text file, copy the contents to another file after removing the blank lines.
2. Find the most frequent words in a text read from a file.
3. Develop a python program to implement the following scenario. Given a give “data.txt” with three columns of data separated by spaces. Read it into 3 separate simple sequences.
4. 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.
5.Write a program to create a new string made of an input string’s first, middle, and last character.
6. Arrange string characters such that lowercase letters should come first.

7. Count all letters, digits, and special symbols from a given string.
8. Find all occurrences of a substring in each string by ignoring the case.
9. Calculate the sum and average of the digits present in a string.
10. Accept 2 matrices and do the given operations.
a.Addition
b.Subtraction
c.Multiplication
11. Create the below Dictionary and Access the value of key 'history'

```
sampleDict= {
   "class":{
"student":{
                                                    "name":"Mike",
                                                        "marks":{
                                                     "physics":70,
                                                      "history":80
                                                            }
                                                            }
                                                            }
   }
```

12. Get the key to a minimum value from the following dictionary
13. Change Brad's salary to 8500 from a given Python dictionary

```
sampleDict = {
                'emp1': {'name': 'Jhon', 'salary': 7500},
               'emp2': {'name': 'Emma', 'salary': 8000},
                'emp3': {'name': 'Brad', 'salary': 6500}
}
```

14. 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
15. Write a Pandas program to calculate the number of characters in each word and convert the first and last character of each word to uppercase of each word in each series.
16. Write a Pandas program to select the rows where the number of attempts in the examination is greater than 2.
Sample DataFrame:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
```

'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

17. Develop a python program to create a dictionary of phone numbers and names of n persons. Display the contents of the dictionary in alphabetical order of their names

18. Develop a python program to implement the following scenario. A book shop details the Title of the book and Number of copies of each title. As books are added to the shop, the number of copies in each should increase and as books are sold the number of copies in each should decrease.

101009/ IT322U
DATABASE MANAGEMENT SYSTEMS LAB

10.1 COURSE INFORMATION SHEET

PROGRAMME: B.TECH- Computer Science & Business systems	**DEGREE:** B.TECH
COURSE: Database Management Systems Lab	**SEMESTER**: III **CREDITS:** 1
COURSE CODE: 101009/ IT322U **REGULATION:** 2021	**COURSE TYPE:** LAB
COURSE AREA/DOMAIN: DATABASE MANAGEMENT SYSTEMS	**CONTACT HOURS:** 3 Practical hours/Week

Preamble:
Database Management Systems Lab course is intended to provide students a hands-on experience in database management concepts. It also provides a strong formal foundation in database concepts, technology, and practice to the students. It gives an exposure to design and develop applications.

Prerequisite:

Code	Course Name	Description	Semester
101009/IT300E	Database Management Systems	Gives concepts of database management systems and exposure to database programming, modeling, and design	3

SYLLABUS:

List of Exercises / Experiments (Experiments No.10, 11 & 13 are not mandatory)
1. Familiarization of Data Definition Language (DDL) and Data Control Language (DCL) commands. 2. Familiarization of Data Manipulation Language (DML) commands. (INSERT, SELECT, DELETE and UPDATE). 3. Implementation of various Aggregate functions and Grouping in SQL. 4. Implementation of Nested Queries. 5. Implementation of Join Queries. 6. Creation of Views. 7. Creation of Stored Procedures and Functions.

8. Exception Handling in SQL.
9. Creation of Triggers and Cursors.
10. Familiarization of Transaction Control Language (TCL) Commands.
11. Familiarization of the NoSQL database using MongoDB.
12. Develop an application to demonstrate database connectivity.
13. Develop a database editor using C language.

CLASS PROJECT (One project per group of at most four members)

Applications like Library Management Systems, Hospital Management Systems, Student Management Systems, Reservation Systems, etc. can be considered project topics.

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T	Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Database System Concepts, McGraw-Hill Education (Asia), Fifth Edition, 2006.
T	Atul Kahate, Introduction to Database Management Systems, *P*earson
T	C. J. Date, A. Kannan and S. Swamynathan, An Introduction to Database Systems, Pearson Education, Eighth Edition, 2009.
T	Patrick O'Neil and Elizabeth O'Neil, Database Principles, Programming and Performance, Harcourt Asia Pte. Ltd., First Edition, 2001.
T	Peter Rob and Carlos Coronel, Database Systems Design, Implementation and Management, Thomson Learning-Course Technology, Seventh Edition, 2007.
T	Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems (7th Edition) , Pearson Education Ltd.
T	Shio Kumar Singh, Database Systems Concepts, Designs and Application, Pearson Education, Second Edition, 2011.

COURSE OBJECTIVES:

1	To provide a hands-on experience in database management concepts
2	To provide a strong formal foundation in database concepts, technology, and practice to the students
3	To present SQL and procedural interfaces to SQL comprehensively
4	To declare and enforce integrity constraints on a database using a state-of-the-art RDBMS

COURSE OUTCOMES:

SNO	DESCRIPTION	Blooms' Taxonomy Level
101009/ IT322U.1	Students will be able to **construct** databases **using** DDL, DCL, and basic DML commands in SQL.	Apply (Level 3), Create (Level 6)
101009/ IT322U.2	Students will be able to **build** nested and join queries.	Apply (Level 3)
101009/ IT322U.3	Students will be able to **apply** procedural SQL concepts like view, exception handling, stored procedure, function, trigger, and cursor in various database applications.	Apply (Level 3)
101009/ IT322U.4	Students will be able to **design** and **develop** database applications.	Create (Level 6)
101009/ IT322U.5	Students will be able to **design** and **develop** a database editor.	Understand (Level 2), create (Level 6)

CO-PO AND CO-PSO MAPPING

	P01	P02	P03	P04	P05	P06	P07	P08	P09	PO10	PO11	PO12	PSO1	PSO2	PSO3

101009/ IT322U.1	2	1	2	-	1	-	-	-	-	-	-	1	1	-	-
101009/ IT322U.2	2	2	1	1	1	-	-	-	-	-	-	-	-	-	-
101009/ IT322U.3	3	2	2	1	-	-	-	-	-	-	-	-	2	1	-
101009/ IT322U.4	2	3	3	2	-	-	-	-	-	-	-	-	2	2	-
101009/ IT322U.5	2	3	3	2	-	-	-	-	-	-	-	-	2	2	2
101009/ IT322U **Overall**	2.2	2.2	2.2	1.5	1	-	-	-	-	-	-	1	1.6	1.67	2

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	**LOW /MEDIUM /HIGH**	**Justification**
101009/ IT322U.1-	M	Understanding the database management system concepts helps to find out solutions to complex

PO1		engineering problems.
101009/ IT322U.1-PO2	L	Knowledge of database design and database models helps to identify, formulate and analyze complex engineering problems reaching substantiated conclusions.
101009/ IT322U.1-PO3	M	This basic knowledge prepares the students for professional careers, with a strong technical foundation.
101009/ IT322U.1-PO5	M	Students will be able to predict and model data management with the help of other modern tools in the field of IT.
101009/ IT322U.1-PO12	L	Students will be able to apply the knowledge in the broadest context of technological change.
101009/ IT322U.1-PSO1	L	Students acquire skills to design and develop databases.
101009/ IT322U.2-PO1	M	Understanding of basic commands helps to solve complex engineering problems.
101009/ IT322U.2-PO2	M	Identify and use appropriate commands to solve complex engineering problems.
101009/ IT322U.2-PO3	L	This knowledge helps to design solutions for complex engineering problems and design system components or processes that meet specified needs.
101009/ IT322U.2-PO4	L	This knowledge prepares the students for professional careers, with a strong technical foundation.
101009/ IT322U.2-PO5	L	Students will be able to use modern tools for query processing.
101009/ IT322U.2-PSO1	L	SQL programming skills can be used for the efficient management of databases.
101009/ IT322U.3-PO1	H	Knowledge of SQL and PL/SQL helps to find out the solution to complex engineering problems.

101009/ IT322U.3-PO2	M	Understanding of DDL, and DML commands in SQL and knowledge of PL/SQL helps to identify, formulate and analyze complex engineering problems reaching substantiated conclusions.
101009/ IT322U.3-PO3	M	Basic knowledge of SQL can be used to design solutions for complex engineering problems and design system components or processes that meet specified needs.
101009/ IT322U .3-PO4	L	This basic understanding helps to use research-based knowledge and research methods including design, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101009/ IT322U.3-PSO1	M	Knowledge about query languages prepares the students for professional careers, with a strong technical foundation.
101009/ IT322U.3-PSO2	L	Database creation and manipulation using SQL helps students to excel in analysing, formulating and solving engineering problems.
101009/ IT322U.4-PO1	M	They understand the basic concepts of cursor and trigger which helps to find out the solution of complex engineering problems.
101009/ IT322U.4-PO2	H	Knowledge on PL/SQL helps to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.
101009/ IT322U.4-PO3	H	Knowledge about concepts like cursor, trigger helps to 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.
101009/ IT322U.4-PO4	M	Understanding of basic database concepts helps in the synthesis of the information to provide valid conclusions.
101009/ IT322U.4-PSO1	M	Understanding of PL/SQL prepares the students for professional careers, with a strong technical foundation in database manipulation.
101009/ IT322U.4-PSO2	M	Database designing and manipulation using Oracle and PL/SQL prepare students to excel in analysing, formulating and solving engineering problems.
101009/	M	They acquire thorough knowledge on database

IT322U.5-PO1		concepts which helps to find out the solution of complex engineering problems.
101009/ IT322U.5-PO2	H	Knowledge on modern databases helps to identify, formulate, review research literature, and analyze complex engineering problems.
101009/ IT322U.5-PO3	H	Data manipulation in recent databases to design solutions for complex engineering problems.
101009/ IT322U.5-PO4	M	Understanding on modern databases 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.
101009/ IT322U.5-PSO1	M	Knowledge on recent databases prepares the students for professional careers, with a strong technical foundation and to nurture the managerial skills to model themselves as entrepreneurs/ key-players in the field of IT.
101009/ IT322U.5-PSO2	M	Knowledge in non-relational databases prepares students to excel in analysing, formulating and solving engineering problems.
101009/ IT322U.5-PSO3	M	Knowledge in recent databases helps in providing IT solutions for different domains.

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

Sl.No.	Description	Proposed Action	Relevance to PO and PSO
1	Oracle database – Storage Organization	Assignment	PO – 2, 3, 4 PSO – 1, 2

PROPOSED ACTIONS: ASSIGNMENT

TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

Sl.No.	Topics	Proposed Action	Relevance to PO and PSO
1	MySQL - Introduction	Assignment	PO – 2, 3, 4 PSO – 1, 2
2	Case Study - Microsoft Access	Assignment	PO – 2, 3, 4 PSO – 1, 2

WEB SOURCE REFERENCES:

1	www.w3schools.com
2	www.sqlcourse.com
3	www.beginner-sql-tutorial.com
4	www.plsql-tutorial.com
5	www.tutorialspoint.com
6	www.cs.rutgers.edu

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

10.2 COURSE PLAN:

SL NO	SCHEDULE	EXPERIMENT
1	WEEK 1	DDL COMMANDS
2	WEEK 2	DML COMMANDS
3	WEEK 3	AGGREGATE FUNCTIONS
4	WEEK 4	GROUPING
5	WEEK 5	NESTED QUERIES
6	WEEK 6	PROCEDURE & FUNCTIONS
7	WEEK 7	EXCEPTION HANDLING

8	WEEK 8	TRIGGER & CURSOR
9	WEEK 9	TCL COMMANDS
10	WEEK 10	COURSE PROJECT
11	WEEK 11	DB EDITOR

10.4 Open Questions

1. Create the following tables:

Customer (CName , CAddress)
CustAccnt (CName , AcctNo)
Account (AcctNo, Balance, Branch_name)
Branch (Branch_name, Branch_city)

a. Create a trigger to ensure that a customer should not have more than 3 accounts.
b. Write a procedure 'Branch_List' that produces for each branch a list of customers and total amount of money held in the accounts of that branch.
c. Display the branch name and customer name of account with the highest balance in the city 'Cochin'.

2. Consider the following database:

Employee(emp-id, job, oldsal, newsal, lowsal, highsal)

Salary hike is given for all employees other than president [use trigger]

See that the following conditions are not violated [Use exceptions]

- lowsal>= newsal<=highsal
- newsal > oldsal
- newsal is only 1.1% greater than the oldsal

3. Create the following tables:

GUEST (Name, Address , Room No, check_in_date, check_out_date)

BILL (Name, Billamount)

ROOM (RoomNo, Charge/day)

HISTORY(Name, Room No, check_out_date)

a) Write a program to print the names of 5 highest billed customers.

b) Write a procedure to generate the bill and store in the BILL table.

c) Write a function to return the maximum bill amount.

d) Create a trigger that inserts values to the table history when a person checks out.

4. Create the following tables:

SalesDetails (Salesman_id (primary key & should start with 'S'), Salesman_name, Quantity, Rate, Date)

SalesCommission (Salesman_id, Commission)

a. Create a trigger called 'give_commission'. Every time when the quantity column is updated then the trigger checks to see if the total amount of a Salesman is over Rs.1000/-. If so give a commission of Rs.200/-

b. Write a function to retrieve the quantity sold by a Salesman. If it is below 50 raise an exception.

c. Display the Salesman name getting the highest commission.

5. Consider the following relations

Salaried_Emp (emp_id, dept, salary)

Retirement (emp_id, service_period, ret_date)

Emp_Details (emp_id, name, dob, street, city, phone)

Do the following operations on these tables:

Calculate the tax for the salaried employees under the following condition.

Salary	Tax
0-5000	no tax
<=8000	5%
<=10,000	10%
Above	15%

Update the salary field after reducing the tax amount from the salary of each employee. Whenever a salaried employee retires(age=55) his details should be recorded in a pension table. His pension amount is half of his salary.

Pension (emp_id, pension_amount, street, city, phone)

6. Write a program to compute the commission. The salesman table records the salesman no., name, commission along with the minimum sales for which the commission is given. If the sales made are greater than the target, he is to be commissioned. The commission amount along with salesman number is to be recorded in commission payable table.

7.) Create a transparent audit system for the table 'ClientMaster'. The system must keep track of records that are being deleted or modified. Table 'ClientMaster' contains fields: client_no, name, balance_due, address. 'AuditClient' contains client_no, name, balance_due, operation, operation_date.

8.) Write a PL/SQL program to list the five highest paid employees.

Employee(emp_no, emp_name, salary)

9.) Write a PL/SQL program to input a dept number and display the number of employees working in a department and also the average salary needed by that department.

Employee(emp_no, emp_name, salary, dept_no)

10.5 ADDITIONAL QUESTIONS

1.Implement a comprehensive e-commerce database management system that handles product catalog, user accounts, shopping carts, orders, and payment processing.
2. Create a database system for a social media platform with user profiles, posts, comments, and connections (friendship/followers).
3. Develop a secure online banking database system that manages customer accounts, transactions, loans, and ATM services.
4. Design a database system for inventory and warehouse management, tracking products, stock levels, orders, and supplier information.
5. Create a hotel management system with features for room bookings, guest check-ins/outs, billing, and room service requests.
6. Build a database system for a ride-sharing platform, including user profiles, ride requests, driver availability, and pricing.

7. Develop a database system for an e-learning platform, managing courses, instructors, student enrollments, and assessments.
8. Create a database system for tourism and travel planning, including tourist attractions, travel itineraries, bookings, and reviews.
9. Design a database system for event management, handling event scheduling, registrations, ticket sales, and attendee management.
10. Build a database system for managing a fleet of vehicles, including vehicle details, maintenance schedules, and driver assignments.
11. Create a database system for agricultural management, tracking crop data, weather conditions, and farming equipment.
12. Develop a database system for hosting gaming tournaments, managing player profiles, brackets, and scores.

101009/EN300F

ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE

11.1 COURSE INFORMATION SHEET 2023-2024

PROGRAMME: COMPUTER SCIENCE & BUSINESS SYSTEMS (2022 ADMISSIONS)	**DEGREE: B.TECH**
COURSE: ESSENCE OF INDIAN TRADITIONAL KNOWLEDGE	**SEMESTER: III CREDITS: NIL**
COURSE CODE: 101009/EN300F **REGULATION: 2021**	**COURSE TYPE: Mandatory Non-Credited Course**
COURSE AREA/DOMAIN: HUMANITIES	**CONTACT HOURS: 2 hours/week – 2 L**

SYLLABUS:.

UNIT 1: Definition of traditional knowledge – nature and characteristics – scope and importance – kinds of traditional knowledge – the contexts in which traditional knowledge develop – the historical impact of social change on traditional knowledge systems; Indigenous Knowledge (IK) – characteristics; Comparison between traditional knowledge and indigenous knowledge, western knowledge, formal knowledge.

UNIT 2: Protection of traditional knowledge – need – significance – value of traditional knowledge in global economy – role of government in protection

UNIT 3: The Scheduled Tribes and Other Traditional Forest Dwellers Act (Recognition of Forest Rights) 2006 – Plant Varieties Protection and Farmer's Rights Act 2001 – Biological Diversity Act 2002 – Protection of Traditional Knowledge Bill 2016.

UNIT 4: Systems of traditional knowledge protection – Legal concepts for the protection of traditional knowledge – Non-IPR mechanisms of traditional knowledge protection – Patents and traditional knowledge – Strategies to increase protection of traditional knowledge – Global legal FORA for increasing protection of Indian Traditional Knowledge.

UNIT 5: Traditional knowledge in science and engineering – in medicine and healthcare – in agriculture, arts and humanities; Importance of conservation and sustainable development – management of biodiversity

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T	Amit Jha, *Traditional Knowledge System in India*, Atlantic Publishers and Distributors Private Limited, New Delhi, 2009.
T	B. Mahadevan, Vinayak Rajat Bhat and Nagendra Pavana R.N., *Introduction to Indian Knowledge System – Concepts and Applications*, PHI Learning Private Limited, New Delhi, 2022.
R	Om Prakash Mishra, *Essence of Indian Traditions*, Khanna Publishers, 2nd edition, New Delhi, 2022.
R	Basanta Kumar Mohanta and Vipin Kumar Singh, *Traditional Knowledge System & Technology in India*, Pratibha Prakashan, New Delhi, 2012.
R	Kapil Kapoor and Awadhesh Kumar Singh, *Indian Knowledge Systems*, D.K. Print World Ltd, New Delhi, 2005.

COURSE PREREQUISITES:

	NIL

COURSE OBJECTIVES:

1	To understand the rich tradition of thought processes, reasoning and inference of the land
2	To understand the significance of traditional knowledge and practices, and the need for their protection
3	Gain insight into India's contribution to development of science and technology
4	Gain familiarity with Indian traditional knowledge in different domains of human activity

COURSE OUTCOMES:

NO	DESCRIPTION
CO1	Differentiate Indian Traditional Knowledge from indigenous, formal and Western knowledge
CO2	Explain the need for protecting traditional knowledge
CO3	Explain the legal and IPR frameworks for protection of traditional knowledge
CO4	Enumerate the contributions and development of Indian traditional knowledge in different fields
CO5	Apply their understanding of traditional knowledge to planning and implementation of developmental activities

MAPPING OF COURSE OUTCOMES TO PROGRAMME 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						
CO 2						2						
CO 3						2						
CO 4						2						2
CO 5	2		2			2	2					2

JUSTIFICATION:

CO	PO	JUSTIFICATION
CO1	PO6	Having clarity on Indian traditional knowledge and practices is essential for developing a contextual perspective
CO2	PO6	Protection of traditional knowledge and utilizing the same for resolving local problems wherever practical is essential for successful professional practice
CO3	PO6	Professionals can play a pivotal role in the protection of traditional knowledge and ensuring the stakeholders benefit from this rich resource
CO4	PO6	Knowing the contributions and development of traditional knowledge in different fields can help the engineer gain a comprehensive perspective on society, which will be useful in developing contextually relevant technologies and solutions
	PO12	Gaining familiarity with India's rich traditional knowledge and practices will set in motion a quest for lifelong learning
CO5	PO1	While resolving engineering problems, the knowledge of traditional systems and practices can provide highly valuable insights
	PO3	Knowledge of the traditional systems and practices can aid in the design and development of contextually and locally relevant solutions
	PO6	In order to successfully work and engage in developmental activities within the Indian context, having the background of traditional knowledge would be helpful
	PO7	To develop environment friendly and sustainable solutions, keeping in mind the contextual factors, knowledge of traditional systems and practices can be very helpful
	PO12	Gaining a historical perspective on how developmental activities have been implemented in the Indian context can be an impetus for engaging in lifelong learning

MAPPING OF COURSE OUTCOMES TO PROGRAMME SPECIFIC OUTCOMES:

	PSO1	PSO2	PSO3
CO1			
CO2			
CO3			
CO4			
CO5			**1**

JUSTIFICATION:

CO	PSO	JUSTIFICATION
CO5	PSO3	An understanding of traditional knowledge in different domains will help in planning and implementing better solutions

WEB RESOURCES:

	LINKS	DESCRIPTION
1	https://www.youtube.com/watch?v=HX7TF5kK4pg	Prof Kapil Kapoor's Lecture on Introduction to Indian Knowledge Systems
2	https://www.youtube.com/watch?v=LZP1StpYEPM&t=6s	Prof. Michael Danino's Lecture on Introduction to Indian Knowledge Systems

TOPICS BEYOND SYLLABUS:

	TOPICS	PROPOSED ACTION
1	Indian philosophical tradition	Student presentation/Discussion

2	Indian approach to psychology	Student presentation/Discussion

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

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

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
Sonia Paul, PhD

Approved by
Dr Anju C.
Head of the Department
Basic Sciences &Humanities

11.2 COURSE PLAN

No	Topic	No. of Lectures
1	**Module1**	
	Introduction	
1.1	Definition of traditional knowledge – nature and characteristics – scope and importance – kinds of traditional knowledge	1
1.2	The contexts in which traditional knowledge develops – the historical impact of social change on traditional knowledge systems	1
1.3	Indigenous Knowledge (IK) – characteristics	1
1.4	Comparison between traditional knowledge and indigenous knowledge, western knowledge, formal knowledge	1
2	**Module 2**	
2.1	Protection of traditional knowledge – need – significance	1
2.2	Value of traditional knowledge in global economy	1
2.3	Role of government in protection	1
3	**Module 3**	
3.1	The Scheduled Tribes and Other Traditional Forest Dwellers Act (Recognition of Forest Rights) 2006	1
3.2	Plant Varieties Protection and Farmer's Rights Act 2001	1
3.3	Biological Diversity Act 2002	1
3.4	Protection of Traditional Knowledge Bill 2016	1
4	**Module 4**	
4.1	Systems of traditional knowledge protection – Legal concepts for the protection of traditional knowledge	1
4.2	Non IPR mechanisms of traditional knowledge protection – Patents and traditional knowledge	1

4.3	Strategies to increase protection of traditional knowledge	1
4.4	Global legal FORA for increasing protection of Indian Traditional Knowledge	1
5	**Module 5**	
5.1	Traditional knowledge in science and engineering	1
5.2	Traditional knowledge in medicine and healthcare	1
5.3	Traditional knowledge in agriculture, arts and humanities	1
5.4	Importance of conservation and sustainable development – management of biodiversity	1

11.3 ASSIGNMENTS 1 & 2

There are two assignments for the course Essence of Indian Traditional Knowledge:

- **Assignment 1 – Student Presentation**
- **Assignment 2 – Research Report**

ASSIGNMENT 1 – STUDENT PRESENTATION

GENERAL INSTRUCTIONS:

Please read the instructions before starting the work on the assignment.

1. Presentations are to be done in the class in the offline mode (using slides/posters/other visual modes as deemed suitable. If required, students may use microphones. They have to arrange this on their own. Ensure visibility, legibility and audibility of your presentation)
2. Each group will be assigned a topic from the table listed below. Every group member has to present a part of this topic.
3. Each group is required to submit a one slide per page printout of the whole presentation (color/black and white) to the teacher in advance.
4. The order of presentation and the split up of the sub topics amongst the group members has to be given in writing to the teacher on the day of the

presentation. Group leaders should coordinate this and provide the list to the teacher.

5. Limit the entire presentation to a maximum of 40 minutes. The rest of the time will be for question and answer and doubt clearing if any. Each individual should thus speak for 4 minutes. (Assign a timekeeper from any of the other groups to help you stay on time)
6. Groups have to arrange their own laptops/devices. The same has to be set up and checked before the session begins. No extra time will be provided for setting up the device or checking during the session. The responsibility of ensuring a professional presentation is completely on the group members. (Advisable to be ready with multiple laptops/back up to meet any contingency)
7. Basic reference materials shall be drawn from the text books/references provided in the syllabus. However, you are encouraged to do additional research using online resources to make the presentation as comprehensive as possible.
8. Marks will be awarded for the thoroughness of your presentation and the additional information you can bring in to the class.
9. Presentations should reflect your comprehension of the topic. Follow the guidelines given herein while preparing your slides:

   Text

   - There should be a heading for each slide
   - Use bulleted points and avoid long sentences/paragraphs - maximum limit is 6 lines per slide and 6 words per line
   - Font size: 30 point for titles, 24 - 28 for text
   - Avoid all capital letters
   - Avoid spelling and grammar mistakes

   Figures and Images

   - Use pictures/photographs wherever relevant. The source should be mentioned in the final slide set apart for giving the references
   - Ensure images are clear
   - All images should be labeled

   General Design

   - Avoid too many colors, clutter or fancy visual effects
   - Use high contrast to ensure visibility: e.g. Black text on white background or black on light background works best
   - Maintain consistency of the same elements on a slide (colors, fonts, styles, placement etc.), as well as, between slides of the full presentation

- Use animation sparingly, if at all. If using transitions, use the same kind throughout
- Edit entire slide deck to ensure organization is logical and design is consistent

10. If the presentation is not done on the scheduled date, it will be summarily awarded zero marks. No requests for change in schedule will be entertained.
11. In case of any clarification required, please contact the teacher. Do not speculate and lose marks.

SCHEDULE OF PRESENTATIONS:

GROUP NUMBER	TOPIC	DATE OF PRESENTATION
1	**India – a historical overview** [To include but not restricted to discussion on the following: Migrations – Invasions – International relations – Tribes – Caste – Urbanization – Women and Transgender in Indian culture Economic – agriculture, industry, trade, taxation Political – kingship, legal system, military Education – seats of learning Spiritual – religions and reformation Social culture – food, clothing, rites of passage, funeral, sports and games etc.]	
2	**Agriculture and Environment** [To include but not restricted to discussion on the following: Early agricultural settlements – influencing factors Crop cultivation – community based environment friendly practices Agricultural practices followed in Kerala Ancient Indian water management and irrigation methods etc.]	
3	**Mathematics & Astronomy** [To include but not restricted to discussion on the following: Concept of time and space – knowledge of the	

	universe Great astronomers and mathematicians of ancient India – their contributions Planetary system and Indian astrology – basic facts – birth charts and predictions etc.]	
4	**Medicine** [To include but not restricted to discussion on the following: Ayurveda, Siddha and Naturopathy – myths and truths Common features – holistic therapeutic approach – natural elements, individual constitution (humors) and the balance recommended Yoga and Meditation Dinacarya and Ritucarya Trigunas and Indian psychology – Pancakosa etc.]	
5	**Engineering & Technology** [To include but not restricted to discussion on the following: Sixty four art forms (Kalas) Vastu-sastra - Architecture – temples, forts, palaces, houses and town planning Metallurgy – mining and ore extraction, coins, traditional metal carving, wootz steel Textile technology – region/culture specific fabric and weaving Dyes and Painting technology Surgical techniques Shipbuilding etc.]	
6	**Indian literary, artistic and cultural tradition** [To include but not restricted to discussion on the following: Sanskrit and Indian languages and dialects Srutis – Smritis – Itihasas – Puranas – Agamas – Darsanas Vedas – Vedangas – Puranas – Upanishads – Ramayana – Mahabharata – Nitisastras – Subhasitas – Arthasastra – Ashtadhyayi	

	Art – Paintings – Music and Musical Instruments – Dance Forms – Sculptures – Sports – Martial Arts – Fairs and Festivals etc.]	
7	**Indian philosophical tradition** [To include but not restricted to discussion on the following: Vedic Schools – Samkhya, Yoga, Nyaya, Vaisesika, Purva-mimamsa, Vedanta/Uttara-mimamsa Jaina and Bauddha philosophy Carvaka Samvad – Jnana – Prama – Pramatr – Prameya – Pramana – Samsaya – Tarka – Avayava etc.]	

NOTE: No request for change in schedule will be entertained

COURSE OUTCOMES: CO1, CO4 and CO5

GRADING RUBRIC FOR ASSIGNMENT 1:

Criteria	**Weightage**	**EXCELLENT**	**GOOD**	**AVERAGE**	**POOR**
Content	**3**	Ample research done and delves into the topic in detail, provides clear purpose and subject, gives pertinent examples, facts, evidences, etc., is able to recognize the relevance of topic	Sufficient research done but fails to convey the content with adequate details and conviction, adequately recognizes the relevance of the topic	Poor research, introduces the topic on a superficial level, gives weak examples, facts, evidences, etc, partially recognizes the relevance of the topic	The content is dealt with vagueness, not substantiated with proper examples, facts, evidences, etc., fails to locate the relevance of the topic
		3	**2**	**1**	**0**

Structure	**2**	Content is well organised along with a good introduction and conclusion	Content is organised adequately along with a good introduction/ conclusion	Content is organised satisfactorily	Content is not properly organised
		2	**1.5**	**1**	**0**
Subject Clarity	**1.5**	Well prepared and delivers ideas in a clear and concise manner without depending too much on notes/ slides	Satisfactorily prepared, attempts to deliver ideas by occasionally depending on notes/ slides	Poor preparation, struggles with the topic, depends too much on notes/ slides	Lacks preparation, reads from notes/ slides directly
		2	**1.5**	**1**	**0**

Presentation Skills	**1**	Maintains proper eye contact with the audience, seldom looking at notes, speaks loudly with proper pronunciation, holds excellent command over language, and exudes confidence through posture & gestures	Limited eye contact, refers to notes, not consistent with pronunciation, good language competence, fails to maintain complete interest of the audience, and maintains good posture with adequate gestures	Minimal eye contact with audience, reads from notes, feeble and monotonous voice, satisfactory language competence, struggles to maintain the interest of the audience, displays mild tension through posture and gestures	Maintains no eye contact, inaudible, lacks proper pronunciation, has poor command over language, lacks confidence and adopts a stiff posture
		1	**0.75**	**0.5**	**0**

ASSIGNMENT 2 – RESEARCH REPORT

Task:

Each student is required to submit a research report on

"Customs and Traditions of my native place, <name of the place**>in Kerala"**

Featuring the following sections:

1. Any regional myth, belief or cultural practice of your native place OR
   Any significance your native place or district has historically/culturally/traditionally etc.
2. Any traditional festival of your native place and its rituals, OR
   Any specific practices followed only in your place/district for common festivals

3. Any two idioms or proverbs in their native tongue connected to seasons, festivals or day to day life with its meaning and significance OR
   Any folk tale which has its origins in your native place/district
4. Any traditional food prepared in your place in accordance with the climate and availability of resources OR
   Any agricultural practice followed/specific crop (or variety) grown only in your district
5. Any place of historical relevance situated in their district OR
   Any literary/artistic/religious/historical personage from your place/district who has made a significant contribution

[Word limit: 300-350 words – approximately 4 written pages of content, excluding the references which have to be given at the end]

Each student is also required to upload the scanned copy of their research report before it has been submitted to the teacher.

GENERAL INSTRUCTIONS:

Please read the instructions before starting the assignment.

1. Submit the assignment on time – hard copy to be submitted on the due date, and soft copy to be submitted before. Late submissions will not be evaluated.
2. Assignment should be handwritten on A4 size plain sheets, with 2 cm margin on all four sides. Don't use pages torn from notebooks for the assignment.
3. Use the Cover Sheet provided. Write your full name, roll number, semester, branch and register number in the space provided.
4. Assignment should reflect your original thinking and understanding of the topic.
5. References should be cited at the end of the report. You may consult family members or other people in your native place for preparing this report. However, verify the authenticity of the content before it is put down.
6. If the assignment is found to be copied/plagiarized, it will be summarily awarded zero marks.
7. In case of any clarification required, please contact the teacher. Do not speculate and lose marks.

COURSE OUTCOMES: CO1, CO4, and CO5

GRADING RUBRIC FOR ASSIGNMENT 1:

Criteria	Weightage	EXCELLENT	GOOD	AVERAGE	POOR
Understanding of the topic	3	Has explored the possibilities of the topic well	Has understood the topic	Some level of understanding of the topic	No understanding of the topic
		3	2	1	0
Research	3	Excellent research	Fair research	Incomplete research	No indication of research
		3	2	1	0
Content & Organization; Grammar & mechanics; Style	1.5	Clearly presents a main idea; Excellent grammar, spelling, vocabulary, syntax and a unique style	There is a main idea; A few errors but attempts to bring in a unique style	Vague idea/; A clear pattern evident in errors and little attempt at style	No main idea/; Errors in plenty and no attempt at style
		1.5	1	0.5	0