Government MCA College

Maninagar (East), Ahmedabad

VISION:

Provide value-based quality education for computer science applications which enable students to solve real-life problems of society.

MISSION:

- To equip our students with good knowledge, skills and attitude to solve real-life problems in the domain of computer applications.
- To establish industry-academia interaction to facilitate the students to work proficiently in the industrial environment.
- To imbibe high moral values and professional ethics.
- To provide a conducive environment so as to achieve excellence in teaching-learning, and research and development activities.

Government MCA College

Maninagar (East), Ahmedabad

Program Objectives (POs):

- PO1 (Foundation Knowledge) : Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.
- PO2 (Problem Analysis) : Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.
- PO3 (Development of Solutions) : Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.
- PO4 (Modern Tool Usage) : Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.
- PO5 (Individual and Teamwork) : Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use

methodologies such as agile.

- PO6 (Project Management and Finance) : Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.
- PO7 (Ethics): Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware
- PO8 (Life-long learning) : Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

Government MCA College

Maninagar (East), Ahmedabad

Program Educational Objectives (PEOs):

- Identify and analyze real-life problems and design computing systems appropriate to their solutions that are technically sound, economically feasible, and socially acceptable.
- Exhibit professionalism, ethical attitude, good communication skills, and teamwork in their profession.
- Adapt to current trends by engaging in life-long learning.

Program Specific Outcomes (PSOs):

- Meet the needs of users within an organizational and societal context through the selection, creation, application, integration, and administration of computing technologies.
- Apply concepts, probability, statistics, and mathematics through calculus (differential and integral), numerical methods, and sciences, including applications appropriate to the field of computing problems.
- Use algorithms, data structures, database management, software design, concepts of programming languages, and computer organization and architecture in computer applications.

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC01094011

Course / Subject Name : Programming with C

w. e. f. Academic Year:	2024-25
Semester:	1
Category of the Course:	Core Course

Prerequisite:	Basic Mathematics and knowledge about number systems
Rationale:	• Learn about the data types, operators and functions in C programming language. • To be able to write code in C programming language for simple problems

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Formulate algorithm/ flowchart for given arithmetic and logical problem.	UN
2	Translate algorithm/flowchart into C program using correct syntax of Operator, conditional & branching statements, iteration and execute it.	AP
3	Write C programs using the concepts of array and functions.	AP
4	Write C programs using the concepts of pointers, structure and union.	AP
5	Develop an application using the concepts file management to solve problems.	CR

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	4	5	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage

1.	Programming Language, Compiler, Interpreter, Liker, Loader, Classification of Programming, Algorithm, Flowchart, Structure of a C Program, First C Program, Comments, C Tokens, Basic data types in C, Variables, Operators & Expression in C, Type conversion and Typecasting.	**7**	**15%**
2.	Basic Screen & Keyboard Input & Output, Introduction to Decision Control Statements, Conditional execution & selection statements, Iterative Statements, Nested Loops, Special Control Statement.	**9**	**20%**
3.	Introduction to Array, One Dimensional Array, Multi-Dimensional Array, Introduction to String, Character & string Functions, Array of String Introduction to User Defined Functions, Using Functions, Function Declaration/Function Prototype, Function Definition, Function call, return statement, Passing Parameters to the function, scope of variables, Storage classes, Recursive Functions, Analysis of Algorithm.	**9**	**20%**
4.	Introduction, Understanding Memory Addresses, Address Operator (&), Introduction to Pointers, void Pointer, Null Pointer, Use of Pointers, Arrays and Pointers, Pointer and String, Pointer Arithmetic, Pointers to Pointers, Array of Pointers, Pointers to an Array, Two-dimensional Arrays and Pointers, Pointers to Functions, Dynamic Memory Allocation. Structure, Structure Declaration, Assessing member of structure, Initialization of Structure, Nesting of Structure, Array of Structure, Array within structure, Structure and Function, Structure and Pointer, Union, Enumeration type, Bit fields.	**12**	**25%**
5.	Using Files in C, Declaration, Opening and Closing of a File, Working with Text Files and Binary Files, Character Input and Output, End of File (EOF), feof() Function, Direct File Input and Output , Files of Records, Random Access to Files of Records, Other File Management Functions: Bitwise Operator (Bitwise AND, Bitwise OR, Bitwise Exclusive-OR, Bitwise NOT, Bitwise Shift Operator), Command-line Arguments, The C Preprocessor, Type Qualifier.	**8**	**20%**
	Total	**45**	**100**

References/Suggested Learning Resources:

Textbook:

1. Pradip Dey, Manas Ghosh, “Programming in C”, 2nd Edition, 2018, Oxford University Press, ISBN: 978-01-9949-147-6.

Reference Books:

1. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316-513-0.
2. Yashavant P. Kanetkar, "Let Us C", 16th Edition, 2019, BPB Publications, ISBN: 978- 93-8728-449-4.
3. Reema Thareja. "Programming in C", 2nd Edition, Oxford University Press.
4. C: The Complete Reference, by Herbert Schildt, Publisher – Tata McGraw Hill.

Suggested Course Practical List:

1. Write a program to that performs as calculator (addition, multiplication, division, subtraction).
2. Write a program to find area of triangle(a=h*b*.5)
   a = area h = height b = base.
3. Write a program to calculate simple interest (i = (p*r*n)/100)
   i = Simple interest p= Principal amount r = Rate of interest n = Number of years.
4. Write a C program to interchange two numbers.
5. Write a C program to enter a distance in to kilometre and convert it in to meter, feet,inches and centimetre.
6. Write a program to compute Fahrenheit from centigrade (f=1.8*c +32).
7. Write a C program to find that the accepted number is Negative, Positive or Zero.
8. Write a program to read marks of a student from keyboard whether the student is pass or fail (using if else)
9. Write a program to read three numbers from keyboard and find out maximum out ofthese three. (nested if else)
10. Write a C program to check whether the entered character is capital, small letter,digit or any special character.
11. Write a program to read marks from keyboard and your program should displayequivalent grade according to following table(if else ladder)

Marks	Grade
100 - 80	Distinction
79 - 60	First Class
59 - 40	Second Class
< 40	Fail

12. Write a c program to prepare pay slip using following data.Da = 10% of basic, Hra = 7.50% of

basic, Ma = 300, Pf = 12.50% of basic, Gross = basic + Da + Hra + Ma, Nt = Gross – Pf.

13. Write a C program to read no 1 to 7 and print relatively day Sunday to Saturday.
14. Write a C program to find out the Maximum and Minimum number from given10 numbers
15. Write a C program to find factorial of a given number.
16. Write a program to reverse a number.
17. Write a program to generate first *n* number of Fibonacci series
18. Write a program to calculate average and total of 5 students for 3 subjects (use nested *for* loops)
19. Read five persons height and weight and count the number of person having height greater than 170 and weight less than 50,
20. Write a program to check whether the given number is prime or not.
21. Write a program to evaluate the series 1^2+2^2+3^2+……+n^2
22. Write a C program to find 1+1/2! +1/3! +1/4! +...... +1/n!
23. Write a program to print following patterns :

```
i  *          ii  *            iii  *****
* *               * *               ****
* * *             * *  *            ***
* * * *           * *  * *          **
* * * * *         * *  ** *         *
```

24. Write a program to print following patterns :

```
i)   1        ii   12345     iii)   55555     iv)  1
     12            1234              4444          22
     123           123                333          333
     1234          12                  22          4444
     12345         1                    1          55555
```

25. Write a program to print following patterns:

```
i)  AAAAA          ii) ABCDE
    BBBB               ABCD
    CCC                ABC
    DD                 AB
    E                  A
```

26. Write a C program to read and store the roll no and marks of 20 students using array.
27. Write a program to find maximum element from 1-Dimensional array.
28. Write a program to sort given array in ascending order.

29. Write a program to replace a character in given string.
30. Write a program to reverse string.
31. Write a program to convert string into upper case.
32. Write a program that defines a function to add first *n* numbers.
33. Write a function in the program to return 1 if number is prime otherwise return 0
34. Write a program to find factorial of a number using recursion.
35. Write a function that will scan a character string passed as an argument and convertall lowercase character into their uppercase equivalents
36. Write a program to read structure elements from keyboard.
37. Define a structure type *struct* personal that would contain person name, joining date and salary using this structure to read this information of 5 people and print the same on screen.
38. Define structure data type called time_struct containing three member's integer hour, integer minute and integer second. Develop a program that would assign values to the individual number and display the time in the following format: 16: 40:51
39. Design a structure student_record to contain name, branch and total marks obtained. Develop a program to read data for 10 students in a class and print them.
40. Write a program to print address of variable using pointer.
41. Write a C program to swap the two values using pointers.
42. Write a C program to print the address of character and the character of string usingpointer.
43. Write a program to access elements using pointer.
44. Write a program to read, print and addition of two Matrices using pointer and user define functions.
45. Write a program for sorting using pointer.
46. Write a program to read n integer number from keyboard and store them into a file All.txt. Read All.txt file, separate even and odd numbers and store them into files Even.txt and Odd.txt respectively and display contents of all the three files.
47. Write a program to accept the contents from the user and store it in the file one line at a time and print the contents of the file.
48. Read a text file which name is given in command line and print the total number of character in each line and total number of lines in a file.
49. Write a program to merge two files into the third file.
50. Program for deleting the spaces from the contents of file.

CO- PO Mapping:

Semester 1	Course Name: Programming with C							
	POs & PSOs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	3	-	-	-	-
CO2	3	3	3	3	-	-	-	-
CO3	3	3	3	3	-	-	-	-
CO4	3	3	3	2	-	-	-	-
CO5	3	3	3	2	-	-	-	-
	3.00	**2.75**	**3.00**	**2.75**	**-**	**-**	**-**	**-**

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC01094021

Course / Subject Name : Fundamental of Computer Organization

w. e. f. Academic Year:	2024-25
Semester:	1
Category of the Course:	Core Course

Prerequisite:	Basic mathematics, introductory computer science, and foundational knowledge of digital electronics.
Rationale:	The "Fundamental of Computer Organization" course is designed to provide students with a comprehensive understanding of the foundational concepts in digital electronics and computer science. These topics are essential for anyone pursuing a career in computer engineering, information technology, or related fields. The rationale behind this course is to equip students with the necessary skills and knowledge to understand and design digital systems, which are the backbone of modern computing devices. By covering number systems and their conversions, basic and advanced logic gates, Boolean algebra, and the basic structure of computers, students will gain a solid foundation that will enable them to understand how computers process, store, and retrieve information. Furthermore, the course delves into the intricate details of register transfer languages, microoperations, and the memory system, which are crucial for understanding the internal workings of a computer. This in-depth knowledge is vital for designing efficient and effective computing systems. Focusing on practical aspects such as Karnaugh maps and memory performance considerations ensures students can apply theoretical concepts to real-world problems. By the end of the course, students will be well-prepared to tackle advanced topics in computer science and engineering, contributing to technological innovations and the development of sophisticated digital solutions.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	perform conversions between different number systems (binary, decimal, octal, hexadecimal) and design basic logic circuits using fundamental logic gates (AND, OR, NOT, NAND, NOR).	R, U, A
02	apply Boolean algebra laws and De Morgan's Theorems to simplify complex logical expressions and design optimized digital circuits using Karnaugh maps.	R, U, A

03	describe the fundamental components of a computer system, including data representation, the basic operational concepts, bus structures, and performance metrics of different computer types.	R, U
04	explain the concepts of register transfer language and microoperations, and describe the execution of instructions, memory-reference instructions, and addressing modes in a computer system.	R, U
05	Explain the various types of memory systems, including RAM, ROM, cache, and virtual memory, and evaluate their performance considerations and the role of secondary storage and RAID in computer architecture.	R, U

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
2	0	2	3	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	UNIT I: NUMBER SYSTEM AND LOGIC GATES Number System: Decimal System, Two-state Devices, Counting in Binary System, Binary Addition and Subtraction, Converting Decimal Number to Binary Numbers, Use of Complements to represent negative numbers in binary and other number systems, Octal and Hexadecimal Number System. Basic Logic Gates: Logic Gates, Logical Multiplication, AND Gate and OR Gate, Complementation and Inverts Evaluation of logical Expression, Evaluation of an Expression containing Parenthesis. NAND Gates and NOR Gates.	4	15
2.	UNIT II FUNDAMENTAL CONCEPTS OF BOOLEAN ALGEBRA: Basic Laws of Boolean Algebra, De Morgan's Theorems, Basic Duality of Boolean Algebra, Derivation of a Boolean Algebra, Interconnecting Gates Sum of Products And Product of Sums, Derivation of POS Expression Derivation of 3 input variables expression	8	25

	Map Simplification Techniques: Karnaugh Maps – 2, 3, 4 Variables, Don't Care Conditions, De Morgan's Theorems, Sum of Products, Product of Sum.		
3.	UNIT III BASIC STRUCTURE OF COMPUTERS : Computer Types, Functional unit, Basic OPERATIONAL concepts, Bus structures, Software, Performance, multiprocessors and multi computers. Data Representation. Fixed Point Representation. Floating – Point Representation. Error Detection codes.	6	20
4.	UNIT III: REGISTER TRANSFER LANGUAGE AND MICROOPERATIONS : Register Transfer language.Register Transfer Bus and memory transfers, Arithmetic Mircrooperatiaons, logic micro operations, shift micro operations, Arithmetic logic shift unit. Instruction codes. Computer Registers Computer instructions – Instruction cycle. Memory – Reference Instructions. Input – Output and Interrupt. Instruction formats. Addressing modes.	8	25
5.	THE MEMORY SYSTEM : Basic concepts semiconductor RAM memories. Read-only memories Cache memories performance considerations, Virtual memories secondary storage. Introduction to RAID.	4	15
	Total	**30**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20	50	30	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Morris Mano, "Computer System Architecture",Third Edition, Pearson Publications
2. P.K. Sinha, "Computer Fundamentals",Sixth Edition, BPB Publications
3. Computer Installation and Servicing By D Balasubramanium | Tata McGraw Hill Education Private Limited
4. Malvino And Leach, "Digital Computer Electronics",Third Edition, Tata McGraw-Hill Education

Logisim is an open-source simulation software (https://logisim.en.softonic.com/) which can be used to perform the simulation experiments.

The proposed list of experiments is as below –

1. **Basic Logic Gates Implementation**
    - Design and simulate basic logic gates (AND, OR, NOT) using Logisim.
    - Verify the truth tables for each gate using different input combinations.
2. **NAND and NOR as Universal Gates**
    - Implement and simulate NAND and NOR gates as universal gates.
    - Construct AND, OR, and NOT gates using only NAND and NOR gates.
3. **Boolean Expression Simplification and Circuit Design**
    - Simplify a given Boolean expression using Boolean algebra.
    - Design and simulate the simplified logic circuit in Logisim.
4. **Karnaugh Map Simplification**
    - Use a Karnaugh map to simplify a 3-variable Boolean expression.
    - Design the simplified circuit in Logisim and verify its functionality.
5. **Half Adder and Full Adder Design**
    - Design and simulate a half adder circuit using basic logic gates.
    - Expand the design to simulate a full adder and test it with all possible input combinations.
6. **4-Bit Binary Adder/Subtractor**
    - Design and simulate a 4-bit binary adder using full adders.
    - Extend the design to create a 4-bit adder/subtractor circuit and simulate it in Logisim.
7. **Multiplexer and Demultiplexer Design**
    - Design and simulate a 4-to-1 multiplexer.
    - Design a 1-to-4 demultiplexer circuit and test its functionality in Logisim.

8. **Design of a 2-Bit Magnitude Comparator**
   - Implement and simulate a 2-bit magnitude comparator using logic gates.
   - Test the circuit with different pairs of binary inputs to verify the comparison results.
9. **Sequential Circuits: Flip-Flops**
   - Design and simulate basic flip-flops (SR, D, JK) using Logisim.
   - Test the behavior of each flip-flop with different input sequences.
10. **4-Bit Binary Counter**
   - Design and simulate a 4-bit binary counter using D flip-flops.
   - Observe and record the counting sequence in Logisim.
11. **Shift Register Design**
   - Implement a 4-bit shift register using D flip-flops.
   - Simulate the shift left and shift right operations in Logisim.
12. **Memory Simulation: RAM Module**
   - Design and simulate a simple RAM module in Logisim.
   - Test read and write operations to the RAM module with different addresses.

CO- PO Mapping:

Semester 1	**Course Name: Fundamental of Computer Organization**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	-	2	-	-	-	-
CO2	3	3	-	-	-	-	-	-
CO3	3	1	-	-	-	-	-	-
CO4	3	2	-	2	-	-	-	-
CO5	2	1	-	-	-	-	-	-
	2.8	**2.25**	**-**	**2**	**-**	**-**	**-**	**-**

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC01094031

Course / Subject Name : Relational Database Management Systems

w. e. f. Academic Year:	2024-25
Semester:	1
Category of the Course:	Core Course

Prerequisite:	Basic knowledge of working with computers
Rationale:	• To understand the fundamental concepts of Database Management Systems and various types of data models. • To understand the concepts necessary for designing, using and implementing database systems and applications • To provide database design approaches using E-R model, EER to Relational Mapping and normalization, Clear understanding for the need of a database and uses the database schema. • To discuss transaction management and concurrency control, and relational management. • To learn various constraints and writing SQL queries.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Describe the core concepts of DBMS & Differentiate various database architectures	Understand
2	Analyze database model and Design relational database using E-R model and UML Classes	Apply
3	Describe functional dependency and Normalize schema relations upto 4NF	Understand, Apply
4	Relate the concept of transaction, concurrency control and recovery in database	Apply
5	Perform DDL and DML SQL queries on schema by enforcing integrity constraints on database	Apply

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC01094031

Course / Subject Name : Relational Database Management Systems

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	P	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Database System** Database and Users: Introduction (Basic Concepts: Data, Database, Database systems, Database Management Systems), Characteristics of Database Approach, Actors on Scene, Workers behind the Scene, Advantages of using the DBMS approach. Database System Concepts and Architecture: Data Models, Schemas, Instances, the three schema architectures and data independence, Database Languages and interfaces, Database System environment, Centralized and client / Server Architecture for DBMS, Classifications of Database Management Systems.	6	15 %
2.	**Entity Relationship Diagram** Using high level conceptual data models for database design (Design Phases of database design), Entity types, Entity Sets, Attributes and keys, Relationship Types, Relationship sets, Roles and structural constraints, Weak entity Types, Refining the ER diagram for company Database, Entity Relationship Diagram Naming conventions Design issues, Example of other Notation: UML class diagram, Relationship types of degree higher than 2 Subclasses, Super Classes, Inheritance Specialization and Generalization Relational Database design by ER and EER to Relational Mapping, Mapping EER model construct to Relations.	11	25 %
3.	**Database Design** Informal Design Guidelines for Relational Schema, Functional Dependencies, Normal Forms based on Primary keys, General definitions of 1NF, 2NF and 3NF, Boyce-Codd Normal Forms (BCNF), Multi-valued Dependency and Fourth Normal Form.	11	20 %

4.	**Transaction processing** Introduction to Transaction Processing Concepts: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, characterizing Schedules based on recoverability and Serializability Relational Model concepts: Relational Model concepts, Relational Model constraints and Relational Database Schemas.	10	25 %
5 (*)	**SQL Concepts**: Basics of SQL, DDL,DML,DCL, structure – creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check, IN operator, Functions - aggregate functions, Built-in functions – numeric, date, string functions, set operations, sub-queries, correlated sub-queries, Use of group by, having, order by, join and its types, Exist, Any, All , view and its types. transaction control commands – Commit, Rollback, Savepoint.	7	15 %
	Total	**45**	**100**

(*): Only Higher order questions / application oriented questions to be asked in the theory exam from Unit 5.

Textbook:

1. Ramez Elmsari,Shamkant B Navathe, “Fundamentals of Database Systems”, Pearson Education, 7th Edition
2. Ivan Bayross, SQL, PL/SQL the Programming Language of Oracle, 4th Edition, BPB Publications

Reference Books:

1. Silberschatz, Korth, Sudarshan, “Database System Concepts”, McGraw Hill Publication. 5th Edition
2. S. K. Singh, “Database Systems : Concepts, Design and Applications”, Pearson Education
3. Peter Rob, Carlos Coronel, “Database Systems : Design, Implementation and Management”, Cengage Learning
4. C. J. Date, A Kannan, S Swaminathan, “An Introduction to Database Systems”, Pearson Education, 8th Edition
5. Steve Suehring, Tim Converse, Joyce Park, PHP 6 and MySQL Bible, Wiley

Suggested Course Practical List:

RDBMS (Module Weightage: 100%)

Tools: Oracle 10g or above

Topics:

1	Data Types
2	Study of DDL Commands (Create , Alter , drop) Table: The Create Table Command, Creating a table from a table (with data, without data, with all columns, with selected columns), Drop Table, Alter Table, Renaming Tables
3	Study DML Commands (Select, insert, update, delete)
4	Constraints: Defining integrity constraints using create table and the alter table command, Dropping integrity constraints in the alter table command
5	Transaction Control statements: Commit, Rollback
6	Advanced Concepts: View, Index, Sequences, rowed, rownum, Default Value Concept
7	Join (Inner Join, Equi Joins, Self Join, Outer Joins)
8	Study subquery concepts
9	Set Operators
10	Study single row functions: String functions, Numeric Functions, Date Functions, Date Conversion Functions
11	Study aggregate / group functions, having
12	Sorting Data, Handling Null values (IS NULL), Like Clause
13	Basic concepts of PL/SQL
14	Create Triggers
15	Data dictionary
16	Procedure, function, package
17	**Desirable** : Security / privileges

Set 1

DEPARTMENT (dept_no, dept_name, location)

1. Create the Simple DEPARTMENT Table.
2. Display structure of department table.
3. Insert below records into Department Table

Dept_no	Dept_name	Location
10	Account	NY
20	HR	NY
30	Production	DL
40	Sales	NY

50	EDP	MU
60	TRG	
110	RND	AH

4. Display all records of Department table
5. Display all department belonging to location 'NY'
6. Display details of Department 10
7. List all department names starting with 'A'
8. List all departments whose number is between 1 and 100
9. Delete 'TRG' department
10. Change department name 'EDP' to 'IT

Set 2

EMPLOYEE (emp_id, emp_name, birth_date, gender, dept_no, address, designation, salary, experience, email)
DEPARTMENT (dept_no, dept_name, location)

Do as directed:

1. Create the EMP Table with all necessary constraints such as In EMP TABLE: Employee id should be primary key, Department no should be Foreign key, employee age (birth_date) should be greater than 18 years, salary should be greater than zero, email should have (@ and dot) sign in address, designation of employee can be “manager”, “clerk”, “leader”, “analyst”, “designer”, “coder”, “tester”.
2. Create DEPT table with neccessary constraint such as
3. Department no should be primary key, department name should be unique.
4. After creation of above tables, modify Employee table by adding the constraints as
5. ‘Male’ or ‘Female’ in gender field and display the structure.
6. Insert proper data (at least 5 appropriate records) in all the tables.
7. Describe the structure of table created
8. List all records of each table in ascending order.
9. Delete the department whose loction is Ahmedabad.
10. Display female employee list
11. Display Departname wise employee Names
12. Find the names of the employee who has salary less than 5000 and greater than 2000.
13. Display the names and the designation of all female employee in descending order.
14. Display the names of all the employees who names starts with ‘A’ ends with ‘A’.
15. Find the name of employee and salary for those who had obtain minimum salary.
16. Add 10% raise in salary of all employees whose department is ‘IT’.
17. Count total number of employees of ‘IT’ department.

18. List all employees who born in the current month.
19. Print the record of employee and dept table as “Employee works in department ‘MBA’.
20. List names of employees who are fresher’s (less than 1 year of experience).
21. List department wise names of employees who has more than 5 years of experience.
22. Crete Sequence to generate department ID
23. List department having no employees

Set 3

Create the following table:
Salesmen table (SNUM, SNAME , CITY , COMMISSION)
Customers (CNUM, CNAME , CITY , RATING , SNUM)
Orders (ONUM, AMOUNT, ODATE, CNUM, SNUM)

SNUM : A unique number assigned to each salesman. SNAME :Thenameofsalesman. CITY :Thelocationof salesmen. COMMISSION: The Salemen's commission on orders

CNUM : A unique number assigned to each customer. CNAME : The name of the customer. CITY : The location of the customer. RATING : A level of preference indicator given to this customer. SNUM : The number of salesman assigned to this customer.

ONUM : A unique number assigned to each order. AMOUNT : The amount of an order. ODATE : The date of an order. CNUM : The number of customer making the order. SNUM : The number of salesman credited with the sale.

Do as directed:

1. Write an Insert script for insertion of rows with substitution variables and insert appropriate data.
2. Produce the order no, amount and date of all orders.
3. Give all the information about all the customers with a specific salesman number.
4. Display the following information in the order of city, sname, snum and commission.
5. List of rating followed by the name of each customer in particular one city e.g. Surat.
6. List of snum of all salesmen with orders in order table without any duplicates.
7. List of all orders for more than certain amount e.g. more than Rs. 1000.
8. List of names and cities of all salesmen in one city e.g. London with commission above 10%.
9. List all customers whose names begins with a letter 'C'.
10. List all customers whose names begins with letter 'A' to'G'.
11. List all orders with zero or NULL amount.
12. Find out the largest orders of salesman from two value e.g. 1002 and 1007.
13. Count all orders of particular date e.g. October 3, 2023
14. Calculate the total amount ordered.

15. Calculate the average amount ordered.
16. Count the no. of salesmen currently having orders.
17. List all salesmen with their % of commission.
18. Assume each salesperson has a 15% commission. Write a query on the order table that will produce the order number, salesman no and the amount of commission for that order
19. Find the highest rating in each city in the form : For the city (city), the highest rating is : (rating)
20. List all in descending order of rating.
21. Calculate the total of orders for each day and place the result in descending order.
22. Show the name of all customers with their salesman's name.
23. List all customers and salesmen who shared a same city.
24. List all orders with the names of their customer and salesman.
25. List all orders by the customers not located in the same city as their salesman.
26. List all customers serviced by salespeople with commission above 15%.
27. Calculate the amount of the salesman commission on each order by a customer with rating above 100.
28. Find all pairs of customers having the same rating without duplication.
29. List all orders that are greater than the average of October 4,2023.
30. Find the average commission of salesmen in London.
31. Find all orders attributed to salesmen in 'London' using both the subquery and join methods.
32. List the commission of all salesmen serving customers in 'London'.
33. Find all customers whose cnum is e.g. 1000 above than the snum of name e.g. Sejal.
34. Count the no. of customers with the rating above than the average of one city e.g. 'Surat'.
35. Find all salesmen with customers located in their cities using ANY and IN.
36. Find all salesmen for whom there are customers that follow them in alphabetical order.
37. Find all customers having rating greater than any customer in particular city e.g. 'Rajkot'.
38. List all orders that has amount greater than atleat one of the orders from 6th October, 2023.
39. Find all orders with amounts smaller than any amount for a customer in 'London'.
40. Find all the customers who have greater rating than every customer in one city e.g. 'Anand'
41. Create a union of two queries that shows the names, cities and ratings of all customers. Those with rating of >=200 should display 'HIGH RATING' and those with < 200 should display 'LOW RATING'.
42. Produce the name and number of each salesman and each customer with more than one current order in the alphabetical order of names.
43. Create union of three queries. First select snum of all salesman in Surat, second, the cnum of all customers in 'Surat' and third, the onum of all orders of 3rd Oct. Retain duplicates between the last two queries but remove the duplicates between either of them and the first.
44. Remove all orders from customer Chirag from the orders table.
45. Set the ratings of all the customers of Piyush to 400.
46. Increase the rating of all customers in Rome by 100.

Set 4

a) Write a PLSQL block which will print Employee list (Empno and Name) EMP (empno, empnm, empadd, salary, date_birth, joindt, deptno)
b) Write a function that returns total number of incomplete jobs, using table JOB (jobid, type_of_job, status)
c) Write a function which displays the number of items whose weight fall between a given ranges for a particular color using table ITEM (itemno, name, color, weight)
d) Write a procedure to display top five highest paid workers who are specialized in "PAINTING" using table WORKER (workerid, name, wage_per_hour, specialized_in, manager_id)

Set 5

Create the database EXAM and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.
APPLICANT (AID, ANAME, ADDR, ABIRTH_DT)
ENTRANCE_TEST (ETID, ETNAME, MAX_SCORE, CUT_SCORE)
ETEST_CENTRE (ETCID, LOCATION, INCHARGE, CAPACITY)
ETEST_DETAILS (AID, ETID, ETCID, ETEST_DT, SCORE)
(This database is for a common entrance test which is being conducted at a number of centers and can be taken by an applicant on any day except holidays)

Do as directed:

1. Modify the APPLICANT table so that every applicant id has an 'A' before its value. E.g. if value is '1123', it should become 'A1123'.
2. Display test center details where no tests were conducted.
3. Display details about applicants who have the same score as that of Ajaykumar in 'ORACLE FUNDAMENTALS'.
4. Display details of applicants who appeared for all tests.
5. Display those tests where no applicant has failed.
6. Display details of entrance test centers which had full attendance between 1st Oct 15 and 15th Oct 16.
7. Display details of the applicants who scored more than the cut score in the tests they appeared in.
8. Display average and maximum score test wise of tests conducted at Mumbai.
9. Display the number of applicants who have appeared for each test, test center wise.
10. Display details about test centers where no tests have been conducted.
11. For tests, which have been conducted between 2-3-17 and 23-4-17, show details of the tests as well as the test centre.
12. How many applicants appeared in the 'ORACLE FUNDAMENTALS' test at
13. Chennai in the month of February?

14. Display details about applicants who appeared for tests in the same month as the month in which they were born.
15. Display the details about APPLICANTS who have scored the highest in each test, test centre wise.
16. Design a read only view, which has details about applicants and the tests that he has appeared for.
17. Write a procedure which will print maximum score centre wise.
18. Write a procedure which will print details of entrance test showing Centre name, candidate id, date and score:
19. Write a trigger which do not allow insertion / updation / deletion of Enterance test details on Sunday.

Set 6

EMP (empno, empnm, empadd, salary, date_birth, joindt, deptno)
DEPT (deptno, deptnm)
Write a PL/SQL block (table above EMP-DEPT table) which takes as input Department name and displays all the employees of this department who has been working since last five years

Set 7

CUSTOMER (cid, fname, lname, city, country, phone)
ORDER (oid, oDate, oNumber, cid, oTotalAmount)

1. List the number of customers in each country. Only include countries with more than 100 customers.
2. List the number of customers in each country, except China, sorted high to low. Only include countries with 5 or more customers.
3. List all customers with average orders between Rs.5000 and Rs.6500.
4. Create a trigger that executes whenever country is updated in CUSTOMER table.
5. Create a function to return customer with maximum orders.
6. Create a procedure to display month names of dates of ORDER table. The month names should be unique.

Set 8

EMPMAST (empno, name, pfno, empbasic, deptno, designation)
DEPT (DNO, DNAME)
Rules: HRA = 15% of basic
DA = 50% of basic
Medical = 100
PF = 8.33%of basic
Print Salary slip. Design your own format

Set 9

Consider the Bank schema as

ACCOUNT (AC_NO, NAME, AC_TYPE, BALANCE_AMT, BALANCE_DATE)
TRANSACTION (AC_NO, DATE, TR_TYPE, AMOUNT, PREV_BALANCE, REMARK)
Note: 1. AC_type may be S for saving or C for current, 2. TR_type may be D for deposit or W for withdrawal.

a. Write a procedure to print the Bank Transaction details by passing from and to dates.

Set 10

Employee (eid, fname, lname, salary)

1. Use a Cursor for Loop inside a function to calculate and return total paid salary to all employees by the company.
2. Modify the function created above to become a procedure and display the total paid salary from the procedure itself. Instead of calculating for all employees, calculate only for those employees whose name starts from a character passed as parameter to the procedure and hence to the cursor.

List of Laboratory/ Active Learning Assignment: If any

Consider Leave Management process includes defining the leave types, assigning entitlements and calculating carry over leaves, employees applying for leaves, managers approving or rejecting the leave requests, importing the leave data into payroll for calculations etc.
Prepare Database design presentation which includes:

1. Purpose of database
2. Find/ Identify Information required for application
3. Conceptual Design
4. Logical Design
5. Physical Design

Apply the concepts learned of conceptual design and logical database design

Learning Resources Required: If Any

1) https://onlinecourses.nptel.ac.in/noc24_cs21/preview
2) https://docs.oracle.com/en/database/index.html
3) https://docs.oracle.com/database/121/SQLRF/toc.htm

Additional Exercises: If Any

NoSQL Database (Desirable)

Tools: MongoDB

1	Introduction, Installation
2	Create Database, Drop Database
3	Create Collection, show collection

4	Insert document, Query Document, Update document, delete document
5	Projection
6	Limiting rows
7	Export and Import

https://docs.mongodb.com/manual/mongo/

CO- PO Mapping:

Semester1	**Relational Database Management System**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	-	-	-	-	-	-	-
CO2	3	3	3	2	-	-	-	-
CO3	3	3	3	3	-	-	-	-
CO4	3	1	2	1	-	-	3	-
CO5	3	3	3	3	-	-	1	-
	3	**2.5**	**2.75**	**2.25**	**-**	**-**	**2**	**-**

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC01094041

Course / Subject Name : Object Oriented Programming using JAVA

w. e. f. Academic Year:	2024-25
Semester:	1
Category of the Course:	Core Course

Prerequisite:	Knowledge of the C programming language and Object Oriented Concept
Rationale:	• To develop proficiency in creating console based applications using the Java Programming Language. • To interpret the concepts of object oriented Programming Language and easily use these concepts in Java program. • To develop application using java.util package (Java data structure). • To implement multi-threaded applications using the Java Programming Language. • To implement application including different file operations.

Course Outcome:

After completion of the course, student will able to:

No.	Course Outcomes	RBT Level*
CO1	Describe Java features, OOP concepts and Write console java program syntax and semantics using datatypes, variables, control and looping statements.	RM, UN, AP
CO2	Explain and Implement class, object, encapsulation and polymorphism in JAVA.	RM, UN, AP
CO3	Apply the concept of inheritance, package and exception handling for reusable, structured and error free programs.	RM, UN, AP
CO4	Use wrapper class and Collection interface to convert and manage list of data.	RM, UN, AP
CO5	Describe and Implement multithreaded programs, Generic class and IO programs.	

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Java** **Java Fundamentals**: The Origins of Java, Java's Lineage: C and C++, How Java Impacted the Internet, Java's Magic: The Bytecode, Moving Beyond Applets, The Java Buzzwords, Object- Oriented Programming (Encapsulation, Polymorphism, Inheritance),Obtaining the Java Development Kit, A First Simple Program, Handling Syntax Errors A Second Simple Program, Another Data Type, Two Control Statements (The if Statement and The for Loop), Create Blocks of Code, Semicolons and Positioning, Indentation Practices, The Java Keywords, Identifiers in Java, The Java Class Libraries Try This 1-1: Converting Gallons to Liters, Try This 1-2: Improving the Gallons-to-Liters Converter **Introducing Data Types and Operators:** Why Data Types Are Important, Java's Primitive Types(Integers, Floating-Point Types, Characters), The Boolean Type, Literals (Hexadecimal, Octal, and Binary Literals, Character Escape Sequences, String Literals), A Closer Look at Variables(Initializing a Variable, Dynamic Initialization), The Scope and Lifetime of Variables, Operators, Arithmetic Operators (Increment and Decrement), Relational and Logical Operators, Short Circuit Logical Operators, The Assignment Operator, Shorthand Assignments, Type Conversion in Assignments, Casting Incompatible Types, Operator Precedence, Expressions (Type Conversion in Expressions, Spacing and Parentheses) Try This 2-1: How Far Away Is the Lightning? Try This 2-2: Display a Truth Table for the Logical Operators **Program Control Statements:** Input Characters from theKeyboard, The if Statement, Nested ifs, The if-else-if Ladder, The switch Statement, Nested switch Statements, The for Loop, Some Variations on the for Loop, Missing Pieces (The Infinite Loop), Loops with No Body, Declaring Loop Control Variables Inside the for Loop, The Enhanced for Loop, The while Loop, The do-while Loop, Usebreak to Exit a Loop, Use break as a Form of goto, Use continue, Nested Loops, Try This 3-1: Start Building a Java Help System,	**7**	**15%**

	Try This 3-2: Improve the Java Help System, Try This 3-3: Finish the Java Help System,		
2	**Class Fundamentals** **Introducing Classes, Objects, and Methods:** Class Fundamentals (The General Form of a Class, Defining a Class), How Objects Are Created, Reference Variables and Assignment, Methods (Adding a Method to the Vehicle Class), Returning from a Method, Returning a Value, Using Parameters (Adding a Parameterized Method to Vehicle), Constructors, Parameterized Constructors, Adding a Constructor to the Vehicle Class, The new Operator Revisited, Garbage Collection, The this Keyword, Try This 4-1: Creating a Help Class **A Closer Look at Methods and Classes:** Controlling Access to Class Members (Java's Access Modifiers), Pass Objects toMethods (How Arguments Are Passed), Returning Objects, Method Overloading, Overloading Constructors, Recursion, Understanding static (Static Blocks), Introducing Nested and Inner Classes, Varargs: Variable-Length Arguments (Varargs Basics, OverloadingVarargs Methods, Varargs and Ambiguity) Try This 6-1: Improving the Queue Class, Try This 6-2: Overloading the Queue Constructor, Try This 6-3: The Quicksort	**10**	**25%**
3	**Inheritance Basics** **Inheritance:** Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call Superclass Constructors, Using super to Access Superclass Members, Creatinga Multilevel Hierarchy, When Are Constructors Executed?, Superclass References and Subclass Objects, Method Overriding, Overridden Methods Support Polymorphism, Why Overridden Methods? (Applying Method Overriding to TwoDShape), Using Abstract Classes, Using final, (final Prevents Overriding, final Prevents Inheritance, Using final with Data Members), The Object Class Try This 7-1: Extending the Vehicle Class **Packages:** Packages (Defining a Package, Finding Packages and CLASSPATH, A Short Package Example), Packages and Member Access (A Package Access Example), Understanding Protected	**10**	**25%**

	Members, Importing Packages, Java's Class Library Is Containedin Packages **Interfaces:** Interfaces, Implementing Interfaces, Using Interface References, Variables in Interfaces, Interfaces Can Be Extended, Default Interface Methods (Default Method Fundamentals, A More Practical Example of a Default Method, Multiple Inheritance Issues), Use static Methods in an Interface, Private Interface Methods, Final Thoughts on Packages and Interfaces, Try This 8-1: Creating a Queue Interface **Exception Handling:** The Exception Hierarchy, Exception Handling Fundamentals (Using try and catch, A Simple Exception Example), The Consequences of an Uncaught Exception(Exceptions Enable You to Handle Errors Gracefully), Using Multiple catch Statements, Catching Subclass Exceptions, Try Blocks Can Be Nested, Throwing an Exception (Rethrowing an Exception), A Closer Look at Throwable, Using finally, Using throws, Three Additional Exception Features, Java's Built-in Exceptions, Creating Exception Subclasses Try This 9-1: Adding Exceptions to the Queue Class		
4	**Autoboxing, java.lang package** Autoboxing, Type Wrappers, Autoboxing Fundamentals, Autoboxing and Methods, Autoboxing/Unboxing Occurs in Expressions **Java.lang package (String, String Buffer, Comparable interface)** **The collection Framework:** Introduction, Collection framework (Collection interface, list interface, set interface, sorted set interface), The collection class, Array list and Link list classes (maintaining the capacity and the link list class **),** iterating elements of collection (the list iterator interface), hash set and tree setclasses	**8**	**15%**
5	**Multi-Threading, Generic, I/O** **Multithreaded Programming** Multithreading Fundamentals, The Thread Class and Runnable Interface, Creating a Thread, (One Improvement and Two Simple Variations), Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities, Synchronization, Using Synchronized Methods, The synchronized Statement Try This 11-1: Extending Thread,	**10**	**20%**

	Try This 11-2: Using the Main Thread, **Generic:** Generics Fundamentals, A Simple Generics Example, A Generic Class with Two Type Parameters, The General Form ofa Generic Class), **Using I/O:** Java's I/O Is Built upon Streams, Byte Streams and Character Streams, The Byte Stream Classes, The Character Stream Classes, The Predefined Streams, Using the Byte Streams (Reading Console Input, Writing Console Output), Reading and Writing Files Using Byte Streams (Inputting from a File, Writing to a File), Automatically Closing a File, Reading and Writing Binary Data		
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	**20**	**70**	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

Textbook:

1. Herbert Schildt, Java™: A Beginner's Guide, 7th Edition
2. Hari Mohan Pandey, JAVA Programming, 978-8131733110, Pearson

Reference Books:

1. Pravin Jain, "The class of Java" Pearson Education, (2010).
2. Paul J. Deitel, HarveyM. Deitel, Java SE8 for Programmers, ISBN: 9789332539068, Pearson
3. Cay S Horstmann, Gary Cornell, "Core Java 2, Volume 1 – Fundamentals",Pearson Education (8th edition – 2008).
4. Ivor Horton's "Beginning Java 2" JDK 5 Edition, Wiley Computer Publishing,(2007).
5. Ken Arnold, James Gosling, David Holmes, "The Java Programming Language",Addison-Wesley Pearson Education (4th Edition – 2005).
6. Raj Kumar Buyya, S. Thamarai Selvi, & Xing Chen Chu, "Object-OrientedProgramming with Java: Essentials & Applications", Tata McGraw Hill
7. Cay Horstmann, "Big Java", Wiley Computer publishing (2nd edition – 2006).
8. Sharan Zakhour, Scott Hommel, Jacob Royal, Isaac Rabinovitch, Tom Risser, Mark Hoeber "The Java Tutorial", Addison-Wesley Pearson Education(4th Edition),

9. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, "The Java Langauge

Suggested Course Practical List:

1	Install the JDK (Download the JDK and install it.) • Set path of the jdk/bin directory. • Create the java program • Compile and run the java program Write a simple "Hello World" java program, compilation, debugging,executing using java compiler and interpreter.
2	Write a program to pass Starting and Ending limit and print all prime numbers and Fibonacci numbers between this ranges.
3	Write a java program to check whethernumber is palindrome or not. Input: 528 Output: It is not palindromenumber Input: 545 Output: It is not palindrome number
4	Write a java program to print value ofx^n. Input: x=5 Input: n=3 Output: 125
5	Write a java program to check Armstrong number. Input: 153 Output:Armstrong number Input: 22 Output: not Armstrong number
6	Write a program in Java to find minimum of three numbers using conditional operator.
7	Write a java program which should display maximum number of given 4 numbers.
8	Write a program in Java to multiply two matrix. Declare a class Matrix where 2D array is declared as instance variable and array shouldbe initialized, within class.
9	Write a java program to create a class "Matrix" that would contain integer values havingvaried Numbers of columns for each row. Print row-wise sum of the integer values for eachrow.
10	Write a Java application which takes several command line arguments, which aresupposed to be names of students and prints output as given below: (Suppose we enter 3 names then output should be asfollows).. Number of arguments = 3 1.First Student Name is = Arun 2.Second Student Name is = Hiren 3.Third Student Name is = Hitesh
11	Write a Java application to count and display frequency of letters and digits from theString given by user as command-line argument.
12	Create a class "Student" that would contain enrollment No, name, and gender andmarks as instance variables and count as static variable which stores the count of the objects; constructors and display(). Implement constructors to initialize instance variables. Also demonstrate constructor chaining. Create objects of class "Student" and displays all values of objects.

13	Write a program in Java to demonstrate use of **this** keyword. Check whether **this** canaccess the Static variables of the class or not. [Refer class student in Q12 to perform the task]
14	Create a class "Rectangle" that would contain length and width as an instance variable and count as a static variable. Define constructors **[constructor overloading (default, parameterized and copy)]** to initialize variables of objects. Define methods to find area and to display variables'value of objects which are created. [Note: define initializer block, static initializer block and the static variable and method. Also demonstrate the sequence of execution of initializer block and static initialize block]
15	Write a java program static block which will be executed before main ( ) method in a Class.
16	Write programs in Java to use Wrapper class of each primitive data types.
17	Write a class "circle" with radius as data member and count the number of instancescreated using default constructor only. [**Constructor Chaining]**
18	Create a class "Vehicle" with instance variable vehicle_type. Inherit the class in a class called "Car" with instance model_type, company name etc. display the information of the vehicle by defining the display() in both super and sub class **[Method Overriding]**
19	Create a class "Account" containing accountNo, and balance as an instance variable. Derive the Account class into two classes named "Savings" and "Current". The"Savings" class should contain instance variable named interestRate, and the "Current"class should contain instance variable called overdraftLimit. Define appropriatemethods for all the classes to enable functionalities to check balance, deposit, andwithdraw amount in Savings and Current account. [Ensure that the Account class cannot be instantiated.]
20	Write a program in Java in which a subclass constructor invokes the constructor of thesuper class and instantiate the values. [refer class Account and sub classes savingAccount and CurrentAccount in Q 19 for this task]
21	Write a program in Java to demonstrate the use of **'final'** keyword in the fielddeclaration. How it is accessed using the objects.
22	Write a java program to illustrate how to access a hidden variable. Class **A** declaresa static variable **x**. The class **B** extends **A** and declares an instance variable **x**. **display( )** method in **B** displays both of these variables.
23	Describe **abstract** class called **Shape** which has three subclasses say **Triangle**, **Rectangle**, **and Circle**. Define one method **area ()** in the abstract class and override this **area ()** in these three subclasses to calculate for specific object i.e. area () of Triangle subclass should calculate area of triangle etc. Same for **Rectangle** and **Circle**
24	Write a java program to implement an interface called Exam with a method Pass (int mark) that returns a boolean. Write another interface called Classify with a method Division (int average) which returns a String. Write a class called Result which implements both Exam and Classify. The Pass method should return true if the mark is greater than or equal to 50 else false. The Division method must return "First" when the parameter average is 60 or more, "Second" when

	average is 50 or more but below 60, “No division” when average is less than 50.
25	Assume that there are two packages, student and exam. A student package contains Student class and the exam package contains Result class. Write a program that generates mark sheet for students.
26	Define a class A in package apack. In class A, three variables are defined of access modifiers protected, private and public. Define class B in package bpack which extends A and write display method which accesses variables of class A. Define class C in package cpack which has one method display() in that create one object of class A and display its variables. Define class ProtectedDemo in package dpack in which write main () method. Create objects of class B and C and class display method for both these objects.
27	Write a java program to implement Generic class Number_1 for both data type int and float in java.
28	Write a java program to accept string to check whether it is in Upper or Lower case. After checking, case will be reversed.
29	Write a java program to use important methods of String class.
30	Write a program in Java to demonstrate use of final class, final variable and final method
31	Write a program in Java to develop user defined exception for 'Divide by Zero' error
32	Write a program in Java to demonstrate throw, throws, finally, multiple try block and multiple catch exception.
33	Write a small application in Java to develop Banking Application in which user deposits the amount Rs 1000.00 and then start withdrawing of Rs 400.00, Rs 300.00 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 500 thereafter.
34	Write a program to write at least 10 objects of the Circle class in a File and to perform basic operations: adding, retrieving, updating, removing elements.
35	Write a program for Java Generics class for Sorting operations: 1. Sorting a list according to natural ordering of elements 2. Reversing sort order 3. Sorting a list whose elements of a custom type 4. Sorting a list using a Comparator. [desirable]
36	Write a program in Java to create, write, modify, read operations on a Text file.
37	Write a java program to illustrate use of standard input stream to read the user input.
38	Write a java program to checks the existence of a specified file.
39	Write a java program to create a file to the specified location.
40	Write a java program to demonstrate the way contents are read from a file.
42	Write a java program to count the availability of text lines in the particular file. A file is read before counting lines of a particular file.

43	Write a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, prime numbers, palindromes).
44	Write a generic method to exchange the positions of two different elements in an array.
	Thread Programming
1	Write a program to implement the concept of threading by extending "Thread" Class.
2	Write a program to implement the concept of threading by implementing "Runnable" Interface.
3.	Write a program that executes two threads. One thread displays "Thread1" every 2,000 milliseconds, and the other displays "Thread2" every 4,000 milliseconds.
4.	Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 50.
5.	Write java program that create and runs following threads: a) print "A" 20 times b) print "B" 30 times c) print "C" 15 times
6.	Write a program in Java to demonstrate use of synchronization of threads when multiple threads are trying to update common variable for "Account" class.
	The collection Framework
1.	Develop a program to create Linked List for "Student" class objects references. "Student" class has std_id, std_name, Array of marks, total_ marks. Calculate total_marks for all students of Linked List. Display Linked List and also display a particular student based on student name as a command line argument.
2.	Develop a program to create Array List for "Employee" class objects references. Employee class has emp_code, emp_name, basic_sal, gross_ sal. Calculate gross_sal for all employees of Array List. Display Array List and also insert an employee object reference in a particular position (input) in Array List. Gross_sal=basic_sal+20% of basic_sal (MA)+30% of basic_sal(HRA)
3.	Develop a program to create Hash Map for "Customer" class objects references. Customer class has Bill_no, cust_mobile_no, Array of item_name, Array of item_unit_price, Array of item_count, total_price. Calculate total_price for all customers of Hash Map. Display Hash Map and also search particular customer's bill based on customer mobile no. Key is mobile no.
4.	Sort "Student" Linked List (mentioned in Q:1) based on std_name using "Comparator" interface.

Active Learning Assignment:

Consider any small application e.g. Library Management System having few transactions like Issue, Return, Renew and Report. Develop Java object oriented application programs and Present the application developed

CO- PO Mapping:

Semester - 1	**Course Name : Object Oriented Programming using JAVA**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	-	-	-	-	-
CO2	3	3	3	-	-	-	-	-
CO3	3	3	3	-	-	-	-	-
CO4	3	3	3	-	-	-	-	-
CO5	3	3	3	-	-	-	-	-
	3	**3**	**3**	**-**	**-**	**-**	**-**	**-**

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code:

Course / Subject Name: Web Technology Project

w. e. f. Academic Year:	2024-25
Semester:	1
Category of the Course:	Core Course

Prerequisite:	Basic computer knowledge.
Rationale:	• The aim of the course is to provide knowledge of web as a tool in presenting information. • Understand basic web languages and its components. • Students can develop a dynamic webpage by the use of HTML, JavaScript and DHTML.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
CO1	Develop web pages using HTML and HTML5 tags and validators.	AP
CO2	Implement client side scripting using Javascript and apply styling using CSS in web pages.	CR
CO3	Develop navigation, events and controlling elements of Angular JS in web pages.	CR
CO4	Design dynamic web applications using PHP.	CR
CO5	Configure Laravel framework and develop database driven PHP web applications	CR

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1	**HTML, XHTML, HTML5 & DHTML** Introduction: HTML,Tags and Attributes, HTML File Structure, Meta, Title, Body Common HTML Tags: Headings, Anchor, Links, Lists, Table, Frames, Forms XHTML: HTML Validator, Block and Inline Elements Introduction to HTML5 DHTML: Introduction to Dynamic Html	6	15
2.	**CSS & JavaScript** Page Designing with CSS: CSS Declarations, Using CSS, Sample CSS File, Selectors, Box Model Concept, Padding, Float and Clear, Z-Index, Websites Benefits of CSS Client-side Scripting: Introduction to JavaScript, Basic Syntax, Variables, Identifiers, DataTypes and Values, Scope, Literals, Reserved Words, Operators and Statements,Functions, Objects (Math, String, Date) Regular Expressions, DOM Event Handling	12	30
3.	**Angular JS:** Single page application: Single page application using AngularJS, Create a module, Define Simple controller Embedding AngularJS script in HTML , AngularJS's routine capability , $routeProvider service from ngRoute, Navigating different pages\,HTML DOM directives, ng-disabled, ng-show, ng-hide, ng-click, Modules (Application, Controller) , Forms (Events, Data validation, ng-click)	8	10
	Introduction to PHP **Introduction**: Basic HTML syntax, Basic PHP Syntax, using SFTP, Testing your script, Sending text to browser, Using the PHP Manual, Sending HTML to the browser, Adding comments to Scripts, Basic debugging steps		

4.	**Variables**: What Are Variables?, Variable Syntax, Types of Variables, Variable Values, Understanding Quotation Marks **Form**: Creating a Simple Form, Choosing a Form Method, Receiving Form Data in PHP, Displaying Errors, Error Reporting, Manually Sending Data to a Page **Numbers**: Creating the Form, Performing Arithmetic, Formatting Numbers, Understanding Precedence, Incrementing and Decrementing a Number, Creating Random Numbers **Strings**: Creating the HTML Form, Concatenating Strings, Handling Newlines, HTML and PHP, Encoding and Decoding Strings, Finding Substrings, Replacing Parts of a String **Control Structures:** Creating the HTML Form, The if Conditional, Validation Functions, Using else, More Operators, Using elseif The Switch Conditional, The for Loop	6	15
5.	**Introduction to LARAVEL:** Why Use a Framework? Why Laravel? Setting Up a Laravel Development Environment, An Introduction to Artisan, Basic Artisan commands, Writing custom artisan commands, calling artisan commands in Normal code, Tinker. **Router and Controllers:** Route Definitions, Route Groups, Views, Controllers, Route Model Binding, Route Caching, Form Method Spoofing, CSRF Protection, Redirects, Aborting the Request, Custom Responses **Blade Template :** Echoing Data, Control Structures (Conditionals, Loops, Or), Template Inheritance, View Composers and Service Injection, Custom Blade Directives Front **End Components:** Elixir, Pagination, Message Bags, String Helpers, Pluralization and localization **Collecting and Validating User Data**: Injecting a Request Object, Route Data, Uploaded Files, Validation, Form Requests, Eloquent Model Mass Assignment, {{versus {!! Auth Controller **Database Eloquent:** configuration, Migration, Seeding, Query Builder, Introduction to Eloquent, Eloquent Events User **Authentication and Authorization :** The User Model and Migration, Using the auth() Global Helper and the Auth Facade	13	30
	Total	**45**	**100**

References/Suggested Learning Resources:

Textbook:

1. M. Srinivasan, Web Technology: Theory and Practice, Pearson India
2. Ivan Bayross, Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP - BPB Publications
3. Jeremy McPeak, Paul Wilton, Beginning JavaScript Wrox Publication
4. Angular From Theory To Practice, Asim Hussain, Version 1.2.0, 2017-11-24
5. Larry Ullman, PHP for Web, fifth Edition, Pearson
6. Matt Stauffer, "LARAVEL Up and Running, A framework for building modern PHP Apps", O'REILLY , 3rd Indian Reprint (ISBN: 978-93-5213-485-4)

Reference Books:

1. Murach's HTML5 & CSS3", Zak Ruvalcaba & Anne Boehm
2. "JavaScript: The Definitive Guide", 6th Edition, David Flanagan, O'Reilly Media
3. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, & Web Graphics", Jennifer Niederst Robbins, O'Rielly
4. "HTML5 Programming with JavaScript", John Paul Mueller, Wiley
5. "HTML5 & CSS3 for the Real World", Estelle Weyl, Louis Lazaris, Alexis Goldstein, Sitepoint
6. Julie C Meloni, "Sams Teach Yourself PHP, MySQL and Apache All in One" 4th edition, Pearson Education
7. Luke Welling, Laura Thomson , PHP and MySQL Web Development — Pearson
8. Mastering Web Application Development with AngularJS, Pawel Kozlowski Peter and Bacon Darwin, Packt Publishing
9. Martin Bean, "Laravel 5 Essentials", Packet Publishing, ISBN 978-1-78528-301-7
10. Fernando Monteiro, "Hands-On Full-Stack Web Development with Angular 6 and Laravel 5 ", Packt Publishing, ISBN 9781788833912

Suggested Course Practical List:

1. HTML Basics: Create a simple HTML document with headings, paragraphs, lists, links, images, and tables.
2. HTML Forms: Design a form with various input types and validate it using HTML5 attributes.
3. CSS Styling: Apply CSS to an HTML document to style text, borders, backgrounds, and layouts using the Box Model.
4. Responsive Design: Create a responsive webpage using CSS media queries.
5. JavaScript Basics: Write a JavaScript script to perform basic arithmetic operations and manipulate the DOM.
6. JavaScript Events: Implement event handling to create interactive elements on a webpage.

7. Form Validation: Create a JavaScript-based form validation system.
8. AngularJS Application: Develop a single-page application (SPA) using AngularJS that includes a module, controller, and view.
9. AngularJS Directives: Use AngularJS directives such as ng-show, ng-hide, and ng-click in a sample application.
10. PHP Basics: Write a simple PHP script to process form data and display it on a webpage.
11. PHP Control Structures: Create a PHP script that uses if-else and switch-case statements for decision-making.
12. Database Connection with PHP: Connect a PHP script to a MySQL database and perform CRUD operations.
13. Laravel Setup: Set up a Laravel development environment and create a simple application using Artisan commands.
14. Laravel Routing and Controllers: Define routes and create controllers in a Laravel application.
15. Laravel Blade Templates: Develop a Laravel application using Blade templates to display dynamic content.

CO- PO Mapping:

Semester	**Course Name : Web Technology Project**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2	-	-	-	-	-
CO2	3	3	3	-	-	-	-	1
CO3	3	3	3	2	-	-	-	-
CO4	3	3	3	2	-	-	-	-
CO5	1	3	3	3	-	-	-	1
	2.6	**2.8**	**2.8**	**2.3**	**-**	**-**	**-**	**1**

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code:

Course / Subject Name: Basic Mathematics

w. e. f. Academic Year:	2024-25
Semester:	1
Category of the Course:	Non-Credit Compulsory Subject for the students who did not have mathematics at undergraduate level.

Prerequisite:	High school mathematics, Basic understanding of algebra and proficiency in problem-solving skills.
Rationale:	This course is designed to equip students with foundational knowledge across various disciplines. Starting with Set Theory & Fundamentals, students learn to manipulate sets and matrices, crucial for applications in database management and algorithms. Mathematical Logic builds on this by teaching rigorous reasoning and logical argument construction, skills vital for software development and formal verification. Integers and Counting deepen students' understanding of concepts like prime numbers and counting techniques, which are essential for cryptography and data science. Relations and Functions introduce students to modeling relationships and mappings applicable to network analysis and machine learning. Lastly, Graphs and Trees provide tools to analyze complex systems, preparing students to solve social network and data structure problems. This structured approach ensures students develop theoretical understanding and practical skills necessary for future academic and professional success in technology-driven fields.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Apply set theory concepts to solve problems involving set operations, describe sets using roster and set-builder notation, and manipulate matrices through various operations.	AP
02	Construct truth tables for logical expressions, apply DeMorgan’s laws to simplify logical statements, and use mathematical induction to prove statements about integers.	AP
03	Calculate greatest common divisors (GCD) and least common multiples (LCM), convert between different numeral systems, and solve counting problems using permutations, combinations, and recurrence relations.	AP
04	Define and classify relations based on their properties, analyze functions for injectivity, surjectivity, and bijectivity, and apply composition and inverse operations to functions.	AP

05	Interpret and construct different types of graphs, analyze paths and cycles within graphs, identify and apply properties of trees, and perform tree traversals using various methods.	AP

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	P	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
0	2	2	0	0	50	50	0	100

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Unit 1: Set Theory & Fundamentals Set Theory** Basic Concepts of Set Theory: Definition, Two Methods to Describe (Represent) Sets; Sets and Subsets, Operations on Sets, Algebraic Properties of Set Operations, The Addition Principle, Sequences: Introduction, Characteristic Functions, Strings and Regular Expressions Matrices: Representation of a Matrix; Equality of Matrices; Special Matrices: Rectangular / Square Matrices, Null (Zero) Matrix, Unit Matrix, Diagonal Matrices, Sum and Difference of 2 Matrices; Multiplication of 2 matrices; Transpose of a Matrix, Symmetric Matrices, Boolean Matrix Operations	5	15
2.	**Unit 2: Mathematical Logic** Introduction, Propositions and Logical Operations, Truth Tables; DeMorgans' Laws; Conditional Statements, Methods of Proof, Mathematical Induction, Mathematical Statements	7	25
3.	**Unit 3: Integers and Counting** Integers: Properties of Integers; Prime Number; Greatest Common Divisor (GCD); Relative Prime; Least Common Multiple (LCM); Representation of Integers in Computer; Decimal, Binary, Octal, and Hexadecimal Representation Counting: Permutations, Combinations, Pigeonhole Principle, Elements of Probability, Recurrence Relations	6	20
4.	**Unit 4: Relations and Functions** Relations: Definition, Binary Relation, Representation, Domain, Range, Universal Relation, Void Relation, Union, Intersection, and	8	25

	Complement Operations on Relations, Properties of Binary Relations in a Set: Reflexive, Symmetric, Transitive, Antisymmetric Relations, Relation Matrix and Graph of a Relation; Partition and Covering of a Set, Equivalence Relation, Equivalence Classes, Compatibility Relation, Maximum Compatibility Block, Composite Relation, Converse of a Relation, Transitive Closure of a Relation R in Set X Functions: Introduction & Definition, Argument. Co-domain, Range, Image, Value of a Function; Examples, Peano's Successor Function; Onto (surjective), Into, One-to-one (injective), Many- to-one, Bijective (one-to-one and onto); examples; Composition of Functions, examples; Inverse Function, Identity Map, Condition of a Function to be Invertible, examples; Inverse of Composite Functions, Properties of Composition of Functions; Binary and n-ary Operations as Mappings (functions), Properties of Binary Operations; Characteristic Function of a Set; Properties, examples; Hashing Functions: Division Method, and Mid-square Method, examples; *Note: Proofs of Theorems are not required*		
5.	**Unit 5: Graphs and Trees** Graphs: Introduction, Definition; Initial & Terminal Nodes, Adjacent Nodes; Directed Edge, Undirected Edge, Directed Graph (Digraph), Undirected Graph, Mixed Graph; Loop (Sling); Distinct Edges, Parallel Edges; Multi-graph, Simple Graph; Weighted Graph; Isolated Nodes, Null Graph; Isomorphic Graphs; In-degree, Out-degree, Total-degree; Subgraphs; Reflexive, Symmetric, Transitive Digraphs; Paths, Length of Path of a Graph; Simple Path (Edge Simple), Elementary Path (Node Simple), Cycle (Circuit), Simple Cycle, Elementary Cycle; Path of Minimum Length (Geodesic), Distance between Two Nodes, Triangle Inequality; Reachability, Reachable Set of a Node, Connected Graphs: Strongly, Unilaterally, Weakly Connected Graphs & Components; Trees: Introduction, Definition, Root, Branch Nodes, Leaf (Terminal Node); Different Representations of Trees; Forests, Subtrees; M-ary Tree, Full or Complete M-ary Tree; Binary Tree, Full (Complete) Binary Tree; Conversion of M-ary Tree to Binary Tree; Traversal of Binary Tree: Pre-order, In-order, and Post-order Traversal *Note: Proofs of Theorems are not required*	4	15
	Total	**30**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
5	**10**	**85**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. J. P. Tremblay and R. Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw-Hill (2010) – only for Unit-5 (Graphs & Trees).
2. Bernard Kolmann & others, "Discrete Mathematical Structure", Pearson Education, Sixth Edition
3. K. H. Rosen, "Discrete Mathematics and its applications", Tata McGraw-Hill, 6th edition
4. D. S. Malik & M. K. Sen, "Discrete Mathematics", Cengage Learning (2004)
5. Edgar G. Goodaire, Michael M. Parmenter. "Discrete Mathematics with Graph Theory", PHI
6. Ralph P Grimaldi & B V Ramana, "Discreet and Combinatorial mathematics: An Applied Introduction", Pearson Education, 5th Edition (2018)
7. J. P. Tremblay and W. K. Grassman. "Logic and Discrete Mathematics", Pearson Education

(b) Open-source software and website:

1. https://www.scilab.org/ (This is an opensource numerical and mathematical simulation tool, students may be given some exercise on using this tool to solve the problems)

Suggested Course Tutorial List:

Following tutorial assignments in each unit should be planned

Unit 1: Set Theory & Fundamentals

1.1 Basic Concepts of Set Theory
- Tutorial Assignments: Solve problems on set operations and properties

1.2 Sequences
- Tutorial Assignments: Analyze given sequences and regular expressions

1.3 Matrices
- Tutorial Assignments: Homework: Solve problems involving matrix operations

Unit 2: Mathematical Logic

2.1 Introduction to Mathematical Logic
- Tutorial Assignments: Solve logic problems using truth tables and proofs

Unit 3: Integers and Counting

3.1 Integers
- Tutorial Assignments: Solve problems on prime numbers and integer representations

3.2 Counting
- Tutorial Assignments: Solve problems involving permutations, combinations, and probability

Unit 4: Relations and Functions

4.1 Relations
- Tutorial Assignments: Solve problems involving binary relations and their properties

4.2 Functions
- Tutorial Assignments: Solve problems on function composition and inversion

Unit 5: Graphs and Trees

5.1 Graphs
- Tutorial Assignments: Solve problems involving graph properties and paths

5.2 Trees

Tutorial Assignments: Solve problems involving tree traversal and representation

Tips for Effective Tutorials

- **Use multimedia:** Incorporate videos, animations, and interactive tools to explain complex concepts.
- **Encourage participation:** Use discussion forums, peer reviews, and group projects to engage students.
- **Provide feedback:** Offer constructive feedback on assignments and activities to help students improve.

Suggested Project List:

Form groups of students. Each group consists of two students. Ask them to write C/C++ program on any two projects from the list given below –

Unit 1: Set Theory & Fundamentals

Project 1: Set Operations
- **Description**: Create a program to perform various set operations (union, intersection, difference, complement) on sets represented as arrays.
- **Key Concepts**: Arrays, functions, set operations.

Project 2: Matrix Operations
- **Description**: Develop a matrix calculator that can perform operations such as addition, subtraction, multiplication, and transpose on matrices.
- **Key Concepts**: 2D arrays, nested loops, functions for matrix operations.

Unit 2: Mathematical Logic

Project 3: Truth Table Generator

- **Description**: Write a program that generates the truth table for given logical expressions.
- **Key Concepts**: Logical operators, loops, conditionals.

Project 4: Proof Verification using Logical Operations

- **Description**: Implement a program to verify simple logical proofs and conditional statements.
- **Key Concepts**: Logical operators, functions, conditionals.

Unit 3: Integers and Counting

Project 5: Number Conversion Tool

- **Description**: Create a tool that converts numbers between different bases (decimal, binary, octal, hexadecimal).
- **Key Concepts**: Loops, conditionals, arrays, string manipulation.

Project 6: Prime Number Finder

- **Description**: Write a program to find all prime numbers up to a given number using the Sieve of Eratosthenes algorithm.
- **Key Concepts**: Arrays, loops, functions.

Project 7: Permutations and Combinations Calculator

- **Description**: Develop a program to calculate permutations and combinations for given values of n and r.
- **Key Concepts**: Recursion, loops, functions, factorial calculation.

Unit 4: Relations and Functions

Project 8: Binary Relation Representation

- **Description**: Implement a program to represent and manipulate binary relations using matrices and graphs.
- **Key Concepts**: 2D arrays, graph representation, matrix operations.

Project 9: Function Composition and Inversion

- **Description**: Create a program to perform composition and inversion of mathematical functions represented as arrays or mappings.
- **Key Concepts**: Arrays, functions, recursion.

Unit 5: Graphs and Trees

Project 10: Graph Traversal Algorithms

- **Description**: Write a program to perform depth-first search (DFS) and breadth-first search (BFS) on a graph.

- **Key Concepts**: Graph representation using adjacency matrix/list, recursion, queues, and stacks.

Project 11: Binary Tree Operations

- **Description**: Implement a binary tree with functionalities such as insertion, deletion, and traversal (in-order, pre-order, post-order).
- **Key Concepts**: Structures, pointers, recursion.

Project 12: Shortest Path Finder

- **Description**: Create a program to find the shortest path in a weighted graph using Dijkstra's algorithm.
- **Key Concepts**: Graph representation, priority queues, algorithms.

CO- PO Mapping:

Semester ___	**Course Name : Basic Mathematics**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	-	-	1	-	-	-	-
CO2	2	-	-	1	-	-	-	-
CO3	2	-	-	1	-	-	-	-
CO4	3	-	-	1	-	-	-	-
CO5	1	-	-	1	-	-	-	-
	1.8	**-**	**-**	**1**	**-**	**-**	**-**	**-**

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC02094011

Course / Subject Name : Data Structure using C

w. e. f. Academic Year:	2024-25
Semester:	2
Category of the Course:	Core Course

Prerequisite:	Proficiency in a programming language
Rationale:	• To develop proficiency in the specification, representation, and implementation of Data Types and Data Structures. • To introduce the concepts of algorithmic paradigms and basic data structures and their applications. • To analyze various algorithms for space and time complexity. • To learn various techniques for representation of the data in nonlinear fashion • To implement and compare various searching and sorting techniques. • To understand basics of graphs and apply appropriate data structures to solve different problems. • To develop a base for advanced computer science study.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Demonstrate familiarity with data structures, algorithm and the storage representation of string & text handing.	U
2	Explain and implement concepts of linear data structures like array, stack, queue, linked list with their representation and perform different operations on them.	AP
3	Explain and implement concepts of tree with their representation and apply various operations on them.	AP
4	Explain and implement concepts of graph with their representation and apply various operations on them.	AP
5	Discuss algorithm & develop programs for Searching and Sorting.	AN

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	Introduction to Data Structure and Algorithm Analysis: Data Structure Definition and classification, Algorithm Analysis, Storage Representation of Strings, Text Handling and KWIC Indexing.	4	10%
2	Linear Data Structures: Arrays, Storage Structure for Arrays, Stack: List Implementation, Applications of Stacks: Function Call, Recursion, Balancing Symbols Queue: List Implementation, Circular Queue, Priority Queue, double ended queue. Linked List: Cursor Implementation, Multi List Applications of Linked List : Addition and Multiplication of Polynomial in one and two variables	12	25%
3	Nonlinear Data Structures: Tree - Basic Tree Concepts, Operations on Binary Trees, Storage Representation & Manipulation of Binary Trees, Conversion of General Tree to Binary Trees, Sequential & Other Representation of Trees, Application of Trees – The Manipulation of Arithmetic Expression, Multi-linked Structures - Sparse Matrices.	10	20%
4	Graphs and Their Representation: Matrix Representation of Graphs, List Structures, Other Representation of Graphs, Breadth First Search and Depth First Search.	7	20%
5	Sorting and Searching Techniques: Sorting – Notation and Concepts, Insertion Sort, Selection Sort, Bubble Sort, Merge Sort, Heap Sort, Quick Sort, Searching - Sequential Searching, Binary Searching, Search Trees – Height Balanced, 2-3 Trees, Weight Balanced Tree, Tree Structures, Hash Table Search Methods, Introduction, Hashing Functions.	12	25%

	Total Hours:	**45**	

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10%	30%	40	20%	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Textbook:

1. Jean-Paul Tremblay, Paul G. Sorenson, "An Introduction to Data Structures with Applications", Tata McGraw-Hill, 2nd Edition, (2007)
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson, Second Edition

Reference Books:

1. Gilberg and Forouzan: "Data Structure- A Pseudo code approach with C" by Thomson publication
2. Data structure in C by Tanenbaum, PHI publication / Pearson publication.
3. Ashok N. Kamthane, "Introduction to Data Structures in C", Pearson Education
4. Cormen, Leiserson, Rivest, Stein,"Introduction to Algorithm", PHI. 2nd Edition
5. Parag H Dave, Himanshu B Dave, "Design and Analysis of Algorithms", Pearson
6. Samir Kumar Bandyopadhyay, Kashi Nath Dey, "Data Structures Using C", Pearson Education, Year: 2004.
7. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education (2002).
8. Horowitz, Sahni, Anderson-Freed, "Fundamentals of Data Structures in C", University Press (2nd edition-2007)

Suggested Course Practical List:

Use C programming language to perform followings Lab work:

1	Create a Structure with following Data Members: 1. Integer Array 2. Size of the Array Sort the Array using various Sorting algorithms such as (i) Selection Sort (ii) Bubble Sort (iii) Two-way Merge Sort (iv) Quick Sort (v) Heap Sort. And store the sorted Array in a text file.

2	Create a Structure with following Data Members: 1. Integer Array 2. Size of the Array Search an element in Array using Linear (Sequential) Search and Binary Search, and Display result in file. For Sequential Search, assume that input array is Unordered and for Binary Search assume that input array is Ordered and develop programs accordingly.
3	Create a "Stack" data structure with following Data members: 1. Integer Array and Stack Pointer (Top of Stack: Is it same as the Size of the Array) Perform the following operations on stack using user-defined functions: Push , Pop , Isempty , Isfull , Peep Create a file which stores all values of Array through Stack. Has it reversed the order of the elements of the Array? Why?
4	Create a "Linked List" structure with the following data members: A Data, A link to the next node Perform the following operations on stack using user-defined functions: 1. Insert a value X at the first place 2. Insert a value X at the end of the list 3. Insert a value X at the place so that it preserves the ordering of the terms in the increasing order. (i) Delete an element whose address is given by X and (ii) Copy a linked linear list Create a file which stores all values of list.
5	Write a program to convert an infix arithmetic expression (parenthesize / unparenthesized) into postfix notation.
6	Write a program to evaluate a postfix expression.
7	Create a structure with the following Data members: 1. Integer Array 2. Size of the Array Search an element in a given list using Binary Search by recursion. And Display result in a file.
8	Create a "Queue" structure with following Data members: 1. Integer Array 2. Size of the Array Perform the following operations on Simple queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of Array.
9	Create a "Queue" user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on Simple queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of list.

10	Create a “Circular Queue” structure with following Data members: 1. Integer Array 2. Size of the Array Perform the following operations on Circular queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of Array.
11	Create a “Circular Queue” user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on Circular queue using user-defined functions: 1. Insert an element 2. Remove an element 3. Display 4. Isfull 5. Isempty Create a file which stores all values of list.
12	Create a user-defined “Linked List” structure with the following data members: 1. A Co-efficient 2. An Exponent 3. A link to the next node Perform the following operations on Singly list using user-defined functions: 1. Create 2. Display 3. Addition 4. Multiplication Create a file which stores all values of list.
13	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on list using user-defined functions: 1. Create a list 2. Traverse the whole list 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.
14	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node Perform the following operations on Circular list using user-defined functions: 1. Create a list 2. Traverse the whole list\ 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.
15	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node 3. A link to the previous node Perform the following operations on the doubly-linked list using user-defined functions: 1. Create a list 2. Traverse the whole list\ 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.
16	Create a user-defined structure with the following data members: 1. A Data 2. A link to the next node 3. A link to the previous node Perform the following operations on doubly-linked Circular list using user defined functions: 1. Create a list 2. Traverse the whole list\ 3. Delete first node 4. Delete last node 5. Delete a node before specified data 6. Insert at first position 7. Insert at last position 8. Insert a node before specified data 9. Insert a node at specified position 10. Count 11. Copy 12. Merge two list 13. Reverse 14. Search 15. Sort Create a file which stores all values of list.

17	Write a program to represent an undirected graph using the adjacency matrix to implement the graph and perform following operations, with menu driven options for following tasks: 1. Create graph 2. Insert an edge 3. Print Adjacency Matrix 4. List all vertices that are adjacent to a specified vertex. 5. Print out vertices using depth first search 6. Print out vertices using breadth first search 7. Exit program
18	Create a user-defined structure with the following data members: 1. A Data, 2. A link to the Left child 3. A link to the Right child 4. Perform the following operations on Binary Search Tree using recursion: 1. Create 2. Traverse (Inorder, Preorder, Postorder) 3. Insert 4. Delete 5. Search 6. Create a file which stores all values of traversal.

List of Laboratory/ Active Learning Assignment: If any

https://www.coursera.org/learn/data-structures-in-c
https://www.coursera.org/specializations/data-structures-algorithms
https://nptel.ac.in/courses/106102064
https://www.mygreatlearning.com/academy/learn-for-free/courses/data-structures-in-c

CO- PO Mapping:

Semester 2	**Course Name : Data Structure using C**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	1	-		-	-	-
CO2	3	3	3	3		-	-	-
CO3	3	3	2	2		-	-	-
CO4	3	3	3	2		-	-	-
CO5	3	3	3	2		-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC02094021

Course / Subject Name : Operating System

w. e. f. Academic Year:	2024-25
Semester:	2
Category of the Course:	Core Course

Prerequisite:	Basic knowledge of computer hardware and software
Rationale:	Understanding of the basic concepts and role of modern Operating systems is an essential for the students. After completion of this course understand the concept of the interaction between user application, OS and hardware architecture. The course will provide an understanding of concept of process, concurrency, deadlock prevention, avoidance, detection and recovery for identification of solution strategies. They will able to understand the issues involved with preemptive vs non-preemptive scheduling. They will be clear about the concept of management of memory, disk and file and efficient use of primary and secondary storage for their application.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Describe the basics of the operating systems and Discuss mechanisms of OS to handle processes & threads	RM
2	Explain the concepts of inter-process communication.	AN
3	Illustrate different conditions for deadlock and their possible solutions	AN
4	Analyze the memory management and its allocation policies	AN
5	Discuss various I/O management techniques and File system	UN

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	UNIT I : INTRODUCTION: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered,Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. PROCESSES, THREAD & PROCESS SCHEDULING: Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads. Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization,Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre- emptive, FCFS, SJF, RR.	12	25
2	UNIT II : INTER-PROCESS COMMUNICATION: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson̓ s Solution, The Producer\ ConsumerProblem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader̓ s & Writer Problem, Dinning Philosopher Problem etc	6	15
3	UNIT III : DEADLOCKS: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker̓ s algorithm, Deadlock detection and Recovery.	5	10
4	UNIT IV : MEMORY MANAGEMENT & VIRTUAL MEMORY: Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, Fixed and variable partition, Internal and External fragmentation and Compaction; Paging: Principle of operation, Page allocation Hardware support for paging, Protection and sharing, Disadvantages of paging. Virtual Memory: Basics of Virtual Memory, Hardware and control structurer, Locality of reference, Page fault , Working Set , Dirty page/Dirty bit, Demand paging, Page Replacement	13	30

	algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).		
5	UNIT V : I/O SYSTEMS, FILE & DISK MANAGEMENT: I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software. File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling algorithms - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks	9	20
	Total Hours:	**45**	

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10%	**30%**	**40%**	**20%**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Textbook:

1. Operating System Concepts Essentials By by Avi Silberschatz, Peter Galvin,Greg Gagne | 9th Edition Wiley Asia Student Edition

Reference Books:

1. Operating Systems Internals and Design Principles By William Stallings | PHI | 5th Edition
2. Operating System: A Design-oriented Approach By Charles Crowley, | 1st Edition - Irwin Publishing
3. Operating Systems: A Modern Perspective By by Gary J. Nutt | Addison-Wesley; 2nd Edition | 2nd Edition
4. Design of the Unix Operating Systems By Maurice Bach, | Prentice-Hall of India | 8th Edition

Suggested Course Practical List:

1) Shell Commands

- date, ls, who, cal, ps, wc, cat, uname, pwd, mkdir, rmdir, cd, cp, rm, mv, diff, chmod,grep, sed, head, tail, cut, paste, sort, find, awk
- Regular expression using sed, grep and awk
- Regular expression using meta characters (^, *, [], (dot), escape, <word>, $, {}, -, +)

2) Task based on commands

1	Write a command to display all lines which begins with "NOKIA" string froman prd.lst file. [File named prd.lst should be created with data such that practical can be performed on it]
2	Delete first, last and all the blank lines from the specified file.
3	Searches for a line which does not start with the vowel letter in any specific file
4	Create a file named customer.dat which contains customer name, address and city. Sort the file in ascending order of customer name and then by city and redirectthe output to name.out file.
5	Create a file named customer.dat which contains customer name, address and city. a) Display all the lines who are started with A to M. b) Display 5 lines after the match of the word "Bombay".
6	Create columnar file named "emp" contains Empno, Empnm, Dept, Salary andHiredate. Write required command for following: a) Cut the Empnm and Salary column and transfer it to empsal file. Display the difference between "emp" and "empsal" file and also display total number of wordsand characters of "emp" and "empsal" file. b) Sort the data in descending order of joining year and also transfer the sorted outputin "experience" file and display only unique lines of "experience" file.
7	Create columnar file named "emp" contains Empno, Empnm, deptno, Desig, andSalary. Write required command for following a. Cut the deptno, Empnm and Salary column and transfer it to empsal file. Alsodisplay first 4 and 7th onwards employees from empsal file.
8	Create columnar file named "emp" contains Empno, Empnm, Desig, and Salary. Writerequired command for following b) Cut the empno,empnm and transfer them into "empinfo.out" file and count frequency of "SHAH" word of that file using single command.
9	Display all such files from your login which has size >= 50.

10	Create columnar file named "restaurants" which contains restaurant name, pizza variety, delivery area and prize columns - Cut the restaurant name and price column and transfer it to file named "economy"and display first 3 three lines of the file "economy".
11	Display all the files with read and write permissions throughout your login and saveoutput in file and also display on terminal
12	Write a command to substitute „/" with „:" throughout the file with all occurrences of /etc/group
13	Using awk command Count number of lines found in a file
14	Using awk command find Sum of fields (Marks).File Format roll#, name, mark1, mark2, mark3

3) Shell Scripts

1	Write a script to compare identically named files in two different directories and ifthey are same, copy one of the min a third directory.
2	Write a script to copy the file system from two directories to anew directory in such away that only the latest file is copied in case there are common files in both the
3	Write a script for generating a mark sheet after reading data from a file.File contains student rollno, name, and marks of three subjects.
4	Write a script to make following file and directory management operations menubased: 1. Display current directory 2. List directory 3. Make directory 4. Change directory 5. Copya file 6. Rename a file
5	Write a script which reads a text file and output the following 1. Count of character, words and lines. 2. File inreverse. 3. Frequency of particular word in the file 4. Lower case letter in place of upper case letter.
6	Write A Script To Perform Following String Operations Using Menu: 1. COMPARE TWO STRINGS. 2. JOINTWO STRINGS. 3. FINDTHE LENGTH OF AGIVENSTRING. 4. OCCURRENCE OFCHARACTER ANDWORDS 5. REVERSETHE STRING.

CO- PO Mapping:

Semester 2	Course Name : Operating System							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	0	0	0	0	0	0	0
CO2	3	2	2	0	0	0	0	0
CO3	3	2	2	0	0	0	0	0
CO4	3	2	2	0	0	0	0	0
CO5	3	2	1	0	0	0	0	0

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC02094031

Course / Subject Name : Computer Networking

w. e. f. Academic Year:	2024-25
Semester:	2
Category of the Course:	Compulsory Subject

Prerequisite:	Basic Concept of Data Structures, Operating systems, Programming Language and Knowledge of Linux OS.
Rationale:	• To equip the students with basics concept of Computer Networks. • To familiarize the students with the standard models for the layered approach OSI and TCP/IP to communication between machines in a network and the protocols of the various layers. • To gain basic insight of programming for network solutions.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Understand the basic concepts of Computer Network, type of network and principle of layering TCP and OSI. (Understand).	Understand
2	Identify the components required to build different types of networks And to understand various Transmission Media wire and wireless for data transfer.	Remember/ Understand
3	Understand different link layer terminologies like error detection and correction techniques. Multiple access protocol and Link layer addressing used in network used for reliable data transfer, flow control and congestion control (Apply).	Understand/ Apply
4	Design network architecture, assign IP addressing and apply various routing algorithms to find shortest paths for network-layer packet delivery. Evaluate error control and flow control at transport layer.	Create / Evaluate
5	Understand and evaluate application layer protocols DNS, HTTP and WWW.	Analyze

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	Introduction: Personal Area Network, Local Area Networks, Metropolitan Area Networks, Wide Area Networks, Internetworks, Network software, protocol hierarchies, Design issues for the layers, connection oriented vs. Connectionless service, service primitives, relationship of services in protocols, Reference Models, Open System Interconnection (OSI), TCP/IP Reference models, Example Networks: ARPANET, NSFNET, Architecture of Internet.	7	15%
2	Physical Layer Guided Transmission Media, Magnetic Media, Twisted Pairs, Coaxial Cable, Power Lines, Fiber Optics, Wireless Transmission, Electromagnetic Spectrum, Radio Transmission, Microwave Transmission, Infrared Transmission, Light Transmission, Digital Modulation and Multiplexing, Public Switched Telephone Networks	7	15%
3	Data Link Layer Design issues, Error detection and correction. Elementary data link protocols: Utopian simplex protocol, a simplex stop and wait protocol for an error-free channel, a simplex stop and wait protocol for noisy channel. Sliding Window protocols, Example data link protocols. Medium Access Control Sublayer The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, IEEE 802.11 Architecture and Protocol Stack, Physical layer, sublayer, Frame Structure, Services, Data link layer switching.	14	30%
4	Network Layer Design issues, Routing algorithms: Optimality principle, shortest path routing, Flooding, distance vector routing, Link State routing, Congestion Control Algorithms, The Network layer in the Internet. Transport Layer: Transport Services, Elements of Transport protocols, Connection establishment, connection release, Error control, flow control, congestion control, UDP and TCP protocols.	13	30%
5	Application Layer: Domain name system (DNS), Electronic Mail; the World Wide Web, HTTP	4	10%
	Total Hours:	**45**	

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20%	**30%**	**20%**	**10%**	**10%**	**10%**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Textbook:

1. Andrew S Tanenbaum, David. J. Wetherall, "Computer Networks", Pearson Education, 5th Edition,
2. Behrouz A. Forouzan, "Data Communications and Networking", Tata McGraw-Hill, Fourth Edition

Reference Books:

1. Computer Networking- A Top-Down approach (6th edition), Kurose and Ross, Pearson
2. Larry L. Peterson, Bruce S. Davie, "Computer Networks: A Systems Approach", Morgan Kaufmann Publishers, Fifth Edition, 2011.
3. Fred Halsall, Computer Networking and the Internet, Addison Wesley, (5th edition)
4. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, "Computer Networks: An Open Source Approach", Mc Graw Hill Publisher, 2011
5. Bhushan H Trivedi, "Computer Networks", Oxford University Press

Suggested Course Practical List:

Network Basics

A. Hardware

1. Install & Test Network Interface Card.
2. Prepare and Test Straight UTP Cable.
3. Prepare and Test Cross UTP Cable.
4. Develop a small Network. (Hands on Training.)

B. Software

5. Install Windows 2003/Windows 2008 Network operating System
6. Install & Configure File Server.
7. Install & Configure Print Server
8. Install & Configure Mail Server
9. Install & Configure Proxy Server

10. Install & Configure Web Server
11. Install & Test Router, Repeater and Bridge.
12. Install a small wireless network

Network Analysis using Wireshark

1. Download Wireshark from its official webpage, it is a network packet analyser https://www.wireshark.org/
2. Install Wireshark under Windows/Linux/MAC/Solaris platform, Windows installer names contain the platform and version. Install WinPcap

Lab1:

Objective: To make the students aware about the IT/Network infrastructure of their parent institute. Introduction to the lab infrastructure, the cabling/cable type and specifications/ switch/ topology/ router/network infrastructure, Internet Connectivity, Wi-Fi Connectivity of your department (labs and lecture halls) and how it is connected with the overall institute level IT infrastructure, How the institute obtains Internet Connectivity from its ISP, The Bandwidth and other specifications, Visit to the Computer Centre of the Institute, Introduction to various Enterprise Servers/Servers of Licensed Software, Server hardware configuration, Server Management Tools, Enterprise Level Firewall, etc. So, lab 1 is intended as an in-campus IT infrastructure industrial visit for students. Students are supposed to prepare a report of this visit. Students are also supposed to note down the model/make of various network interface devices (NIC card, switch, router) used in their lab/department/institute, download their technical specifications from their respective web-site and attach those specifications in the report. The significance and interpretation of these specifications may be discussed by the faculty member during the corresponding lab/lecture hours pertaining to that particular device.

Lab 2:
Objective: To make the students aware about and learn the detailed use of the following OS level TCP/IP diagnostic and troubleshooting commands: ping, ns lookup

Lab 3:
Objective: To make the students aware about and learn the detailed use of the following OS level TCP/IP diagnostic and troubleshooting commands: ipconfig, arp, netstat, tracert, telnet

Lab 4a:
Objective: Implement a simple TCP socket based client server program in Python in which the client connects to the server. The server displays the ip address and port number of client and sends an acknowledgement message back to client. The client displays the received acknowledgement message on screen.

Lab 4b:
Objective: Implement a simple UDP socket based client server program in Python in which the client connects to the server. The server displays the ip address and port number of client and sends an acknowledgement message back to client. The client displays the received acknowledgement message on screen.

Lab 5:
Objective: Wireshark Installation, understanding the Wireshark Environment / Menu System, to start and stop live capture of traffic from given wired Ethernet network interface, capturing options, store captured data in different supported file formats, to open already stored captured data file.

Lab 6:
Objective: Learn and use view level filters and capture level filters in Wireshark for different traffic types like Ethernet, ARP, IP, TCP, UDP, DNS, HTTP, etc. For each captured category, observe how different headers are encapsulated within each other. E.g. TCP encapsulated within IP, HTTP encapsulated within TCP, etc.

Lab 7:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header and payload: Ethernet and ARP

Lab 8:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header and payload: IP and ICMP

Lab 9:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header and payload: TCP and UDP

Lab 10:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header and payload: HTTP and DNS

Lab 11:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header and payload: FTP, SMTP and Telnet

Lab 12:
Objective: Capture the following traffic types and Interpret/ Analyze the corresponding header and payload: Ethernet and ARP

Lab 13:

Objective: Capture Wi-Fi and Bluetooth Traffic and Interpret/ Analyze the corresponding header and payload using Wireless Traffic Sniffing tools like Wireshark-USB/AirCrackng/ Kismet, etc.

Lab 14:
Objective: Analyze Email Traffic: Normal POP Communications, POP Problems, Dissect the POP Packet Structure, Filter on POP Traffic, Normal SMTP Communications, SMTP Problems, Dissect the SMTP Packet Structure, Filter on SMTP Traffic

Lab 15:
Objective: Analyze IEEE 802.11 (WLAN): Wireless LANs (WLANs) Traffic, Signal Strength and Interference, Capture WLAN Traffic, 802.11 Traffic Basics like Data Frame, Normal 802.11 Communications

The Practical Examination Exercises would be based on Lab Exercises based on above list EXCEPT Lab # 1, 4a, 4b, 13 and 15. For Lab # 1, 4a, 4b, 13 and 15, documentation/report/file should be maintained.

References:
1) https://technet.microsoft.com/en-us/library/bb727023.aspx
2) https://docs.python.org/2/howto/sockets.html
3) https://www.aircrack-ng.org/
4) https://www.kismetwireless.net/

Note: Some of the practical's form the above practical list may have seemingly similar definitions. For better learning and good practice, it is advised that students do maximum number of practical's. In the practical examination, the definition asked need not have the same wordings as given in the practical list. However, the definitions asked in the exams will be similar to the ones given in the practical list.

Exploring using Python

1. Implement a Python Program to print host name and IP address of local host.
2. Implement a Python Program to print host name and IP address of remote host where IP address of remote host is available.
3. Implement a Python Program to print host name and IP address of remote host where hostname of remote host is available.
4. Implement a TCP port scanner program in python for local host.
   (Note: Do not try this program for a remote host, especially outside your domain. It could cause legal problems)
5. Implement a UDP port scanner program in python for local host.
   (Note: Do not try this program for a remote host, especially outside your domain. It could Cause Legal problems)
6. Implement a TCP based client server program in python using TCP sockets where Server displays the following: a) Host Name, IP address and Port Number on which it is hosted b) IP

address and port number of a client requesting connection. Server sends the message "Thanks for Connecting!" back to client. Client displays this message on screen.

7. Implement a UDP based client server program in python using UDP sockets where Server displays the following:
   a) Host Name, IP address and Port Number on which it is hosted
   b) IP address and port number of a client sending some dummy message. Server displays the dummy message on screen. Server sends the message "Thanks for Message!" back to client. Client displays this message on screen.
8. Implement a TCP based echo client server program in python.
9. Implement a UDP based echo client server program in python.
10. Implement a TCP based daytime client server program in python.
11. Implement a UDP based daytime client server program in python.
12. Implement a TCP based client server text chat program in python.
13. Implement a UDP based client server text chat program in python.
14. Implement a TCP based echo client server program in python with a multi-threaded server.
15. Implement a TCP based daytime client server program in python with a multi- threaded server.
16. Implement a web client using urllib to:
    a) Display the html source of a given URL on screen
    b) Display the URL visited
    c) Display the header information transmitted in the http response sent by the contacted web-site/web-server.
    d) Display the http server status code
17. Implement an ftp client using ftplib which connects to an ftp server, takes login/password from user, displays directory list and permits upload and download of files.
18. Write a Python program that makes a connection to a web server and retrieve/display a document using the HTTP protocol.
19. Write a Python program that makes a connection to a web server and retrieve an image using the HTTP protocol.
20. Write a python program to implement a simple server-client program :
21. Write a python program to implement socket programming using multi-threading

Reference Material for Python Network Programming:

22. Learning Python Network Programming, Dr. M.O. Faruque Sarker & Sam Washington, Packt Publishing, 2015
23. Python Network Programming Cookbook, 2nd Edition, Pradeeban Kathiravelu, Dr. M.O. Faruque Sarker, Packt Publishing, 2017
24. Foundations of Python Network Programming, 3rd edition, Brandon Rhodes & John Goerzen, Apress Publications

25. Beginning Python: From Novice to Professional, 2nd edition, 2009 (Indian Reprint), Apress Publications
26. https://docs.python.org/3/howto/sockets.html
27. https://docs.python.org/3/library/socket.html

CO- PO Mapping:

Semester 2	Course Name: Computer Networking							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	-	-	1	-	-	-	-
CO2	2	-	-	1	-	-	-	-
CO3	3	1	2	2	-	-	-	-
CO4	3	1	2	2	-	-	-	-
CO5	2	-	-	1	-	-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC02094041

Course / Subject Name : Full Stack Development

w. e. f. Academic Year:	2024-25
Semester:	2
Category of the Course:	Core Courses

Prerequisite:	Fundamental knowledge of computer networking and internet concepts.
Rationale:	• To design and develop a robust application • To understand the fundamentals of web programming and client-side scripting. • To learn web server and server-side development using NodeJS. • To understand API development with Express Framework. • To understand and architect databases using NoSQL. • To learn the advanced client-side scripting and ReactJS framework and app implementation in Cloud.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	To become knowledgeable about the most recent web development technologies and to write client-side scripting HTML, CSS and Ajax.	Remember and Understand
2	Implement and architect the server side of the web application, javascript and learn and implement express framework, learn and apply JSON.	Understand/ Apply
3	Learn and implement advanced Node JS and Database, Architect NoSQL databases with MongoDB.	Understand/Apply
4	To learn the advanced client-side scripting using ReactJS, fetch through API .	Analyze
5	Create app implementation in Cloud and evaluate. Implement a full-stack Single Page Application using React, NodeJS and MongoDB and deploy on Cloud.	Create/Evaluate

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Explain principles of client-server communication and Design interactive webpages using HTML, CSS, and javascript integrated with AJAX.	RM,UN,AP

2	Describe and Utilize Node.js and NPM to develop server side applications, serve files with the http module, and implement APIs to fetch JSON data using Express framework.	RM, UN, AP
3	Explain and develop full-stack applications that integrate Node.js with both SQL and NoSQL databases for data handling and operations.	RM, UN, AP
4	Explain principles of React, including ReactDOM, JSX, and their properties and Develop dynamic and responsive user interfaces by managing state, utilizing local storage, handling events and component communication in React applications.	RM, UN, AP
5	Discuss services provided by various cloud providers, benefits of Virtual Private Cloud and deploy containerized application at scale using Kubernetes.	RM,UN,AP,AN,EV

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	Introduction to Web: Server - Client - Communication Protocol (HTTP) – Structure of HTML Documents – Basic Markup tags – Working with Text and Images with CSS– CSS Selectors – CSS Flexbox - JavaScript: Data Types and Variables - Functions - Events – AJAX: GET and POST.	8	15%
2	Introduction to Web Servers – Javascript in the Desktop with NodeJS – NPM – Serving files with the http module – Introduction to the Express framework – Server-side rendering with Templating Engines – Static Files - async/await - Fetching JSON from Express.	9	20%

3	Advanced Node JS and Database: Introduction to NoSQL databases – MongoDB system overview - Basic querying with MongoDB shell – Request body parsing in Express – NodeJS MongoDB connection – Adding and retrieving data to MongoDB from NodeJS – Handling SQL databases from NodeJS – Handling Cookies in NodeJS – Handling User Authentication with NodeJS.	10	25%
4	Advanced Client Side Programming: React JS: ReactDOM - JSX - Components - Properties – Fetch API - State and Lifecycle - -JS Localstorage - Events - Lifting State Up - Composition and Inheritance.	10	25%
5	App Implementation in Cloud: Cloud providers Overview – Virtual Private Cloud – Scaling (Horizontal and Vertical) – Virtual Machines, Ethernet and Switches – Docker Container – Kubernetes.	8	15%
	Total Hours:	**45**	

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10%	20%	30%	10%	10%	20%

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Textbook:

1. Internet and World Wide Web:How to Program, Paul Deitel , Harvey Deitel & Abbey Deitel, Pearson Education, 5th Edition, 2018.
2. HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery), DT Editorial Services, Dreamtech Press, 2nd Edition, 2016.

Reference Books:

1. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer, Apress Publications, 1st Edition, 2018.
2. Laura Lemay, Rafe Colburn & Jennifer Kyrnin, Mastering HTML, CSS & Javascript Web Publishing, BPB Publications, 1st Edition, 2016.

3. Matt Frisbie, "Professional JavaScript for Web Developers", Wiley Publishing, Inc, 4th Edition, ISBN: 978-1-119-36656-0, 2019
4. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by Step Guide to CreatingDynamic Websites by Robin Nixon
5. Alex Giamas, Mastering MongoDB 3.x, Packt Publishing Limited, First Edition, 2017.
6. Alex Banks, Eve Porcello, "Learning React", O'Reilly Media, Inc, 2nd Edition, 2020
7. Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB. Copyright © 2015 Byazat Mardan
8. Mastering Full Stack React Web Development Paperback – April 28, 2017 by TomaszDyl , Kamil Przeorski , Maciej Czarnecki

Suggested Course Practical List:

1. Write a program to create a website using HTML CSS and JavaScript.
2. Write a program to build a Chat module using HTML CSS and JavaScript.
3. Build an online MCQ quiz app. The questions and options should be fetched based on the chosen topic from a NodeJS server. The questions can be stored in a JSON file in the backend. Once the user has answered the questions, the frontend must send the chosen options to the backend and the backend must identify the right answers and send the score back to the front end. The frontend must display the score in a separate neatly designed page.
4. Build a blog website where you can add blog posts through a simple admin panel and the users can view the blog posts. The contents of the blog posts can be stored in either MongoDB or MySQL database. The home page should contain the titles of the blog post and the full post can be viewed by clicking the title. Frontend can be built either using React or through template engines served by the NodeJS server.
5. Take Amazon ecommerce or Facebook social media website/app. Analyze what the API endpoints would have been used for and how the frontend interacts with the backend. The networks tab in the browser's developer tools can be used if required.
6. Architect an entire database structure for an E-Commerce application for MongoDB. Discuss how the database would have been structured if you were using a SQL database.
7. Create a Simple Login form using React JS
8. Build a robust Todo List App with ReactJS.
9. Write a program to create a voting application using React JS
10. Build a simple calculator app with React. The user should be able to add numbers and operations to the app by clicking on buttons, just like you would do in a mobile phone. The moment the operation and the two operations are defined, the answer should be displayed
11. Create a project on Grocery delivery application

List of Laboratory/ Active Learning Assignment:

https://www.coursera.org/learn/the-full-stack

https://github.com/topics/full-stack-web-development?l=html
https://www.upgrad.com/blog/skills-to-become-a-full-stack-developer/
https://www.upgrad.com/blog/full-stack-coding-project-ideas-topics-for-beginners/

Additional Exercises:

1. Write a program to create and Build a Password Strength Check using Jquery
2. Write a program to create and Build a star rating system using Jquery
3. Create a blog using React JS

CO- PO Mapping:

Semester 2	**Course Name : Full Stack Development**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	0	0	0	0
CO2	3	3	3	3	0	0	0	0
CO3	3	3	3	3	0	0	0	0
CO4	3	3	3	3	0	0	0	0
CO5	3	3	3	3	2	0	0	0

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC02094051

Course / Subject Name : Programming with Python

w. e. f. Academic Year:	2024-25
Semester:	2
Category of the Course:	Core Courses

Prerequisite: Basic Concept of Programming

Rationale:	• To develop proficiency in creating applications using the Python Programming Language. • To be able to understand the various data structures available in Python Programming language and apply them in solving computational problems. • To be able to draw various kinds of data visualization techniques using PyLab, matplotlib. • To be able to understand the DB API in Python.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Demonstrate programs using simple Python statements, expressions, control flow, lists, tuples & dictionaries	AP
2	Explain and implement functions, files, exception, modules and packages concept in Python for solving problems.	AP
3	Demonstrate programs using object oriented concepts.	AP
4	Demonstrate use of data visualization techniques using PyLab, matplotlib.	AP
5	Demonstrate use of Regular expressions in python programming.	AP
6	Demonstrate use of database connectivity in Python.	AP

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	**Introduction to Python** The basic elements of Python, Objects, expressions and numerical Types, Variables and assignments, IDLE Branching programs, Strings and Input, Iteration Structured Types, Mutability and Higher-order Functions: Tuples, Ranges, Lists and Mutability (Cloning and list comprehension), Strings, Tuples and Lists, Dictionaries	7	15%
2	**Functions, Exception, Modules and Files** **Functions:** Difference between a Function and a Method, Defining aFunction, Calling a Function, Returning Results from a Function, Returning Multiple Values from a Function, Functions are First Class Objects, Pass by Object Reference, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Local and Global Variables, The Global Keyword, Passing a Group of Elements to a Function, Recursive Functions, Anonymous Functions or Lambdas (Using Lambdas with filter() Function, Using Lambdas with map() Function, Using Lambdas with reduce() Function), Function Decorators, Generators, Structured Programming, Creating our Own Modules in Python, The Special Variable_name **Exceptions**: Errors in a Python Program (Compile-Time Errors, Runtime Errors, Logical Errors), Exceptions, Exception Handling, Types of Exceptions, The Except Block, the assert Statement, User-Defined Exceptions, Logging the Exceptions **Files:** Files, Types of Files in Python, Opening a File, Closing a File, Working with Text Files Containing Strings, Knowing Whether a File Exists or Not, Working with Binary Files, The with Statement,Pickle in Python, The seek() and tell() Methods	10	20%
3	**Classes and Object-oriented Programming** Abstract Data Types and classes, Inheritance, Encapsulation and Information hiding Mortgages and Extended ExamplesCase Study: Banking Application	10	20%
4	**Advanced Topics I: Data Science and Data Visualization using Python**	8	20%

	Data Science Using Python: Data Frame (Creating Data Frame from an Excel Spreadsheet, Creating Data Frame from .csv Files, Creating Data Frame from a Python Dictionary, Creating Data from Python List of Tuples, Operations on Data Frames) **Data Visualization:** Bar Graph, Histogram, creating a Pie Chart, Creating Line Graph **Plotting:** Plotting using PyLab, Plotting mortgages and extended examples		
5	**Advanced Topics II: Regular Expressions** **REs and Python:** Regular Expressions, Sequence Characters in Regular Expressions, Quantifiers in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expressions on Files, Retrieving Information from a HTML File Case Study: Screen Scrapping	**4**	**15%**
6	**Python's Database Connectivity** Verifying the MySQL dB Interface Installation, Working with MySQL Database, Using MySQL from Python, Retrieving All Rows from a Table, Inserting Rows into a Table, Deleting Rows from a Table, Updating Rows in a Table, Creating Database Tables through Python	**3**	**10%**
	Total Hours:	**42**	

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
15	**25**	**60**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

1. Text Book(s):

1. John V Guttag. "Introduction to Computation and Programming Using Python", 2nd Edition, Prentice Hall of India
2. R Nageswara Rao, Core Python Programming, 2nd Edition, Dreamtech Press

2. Reference Books:

1) Wesley J Chun, Core Python Applications Programming, 3rd Edition. Pearson
2) Luke Sneeringer, Professional Python, WROX
3) Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python, Pearson
4) Doug Hellmann, The python 3 standard Library by example, Pearson Education
5) Alex Martelli, Python Cookbook, O'REILLY
6) Laura Cassell, Python Projects, WROX
7) Daniel Y Chen, Pandas for Everyone: Python Data Analysis, 1st Edition, Pearson Eduction

Web Resources:

1) Charles Severance, Python for informatics: www.pythonlearn.com
2) Swaroop C H. "A Byte of Python", http://www.swaroopch.com/notes/python
3) "Python Programming", http://en.wikibooks.org/wiki/Python_Programming
4) "The Python Tutorial", http://docs.python.org/release/3.0.1/tutorial/
5) "Learn Python the Hard way", http://learnpythonthehardway.org/
6) Dive into Python 3: http://www.diveintopython.net/

Practical List

Tools: Python 3.x, IDLE

Part I: Core Python

- **Basics**

1	Write a Python Program to Convert Celsius to Fahrenheit and vice –a-versa.
2	Write a program in python to swap two variables without using temporary variable.
3	Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal
4	Write a program to make a simple calculator (using functions).
5	Write a program in python to find out maximum and minimum number out of three user entered number.
6	Write a program which will allow user to enter 10 numbers and display largest oddnumber from them. It will display appropriate message in case if no odd number is found.
7	Write a Python program to check if the number provided by the user is an Armstrong number.

8	Write a Python program to check if the number provided by the user is a palindrome or not.
9	Write a Python program to perform following operation on given string input: a) Count Number of Vowel in given string b) Count Length of string (do not use Len ()) c) Reverse string d) Find and replace operation e) check whether string entered is a palindrome or not
10	Define a procedure histogram () that takes a list of integers and prints a histogram to thescreen. For example, histogram ([4, 9, 7]) should print the following: **** ********* *******
11	Write a program in python to implement Fibonacci series up to user entered number. (Use recursive Function)
12	Write a program in python to implement Factorial series up to user entered number. (Use recursive Function)
13	Write a program in Python to implement readline, readlines, write line and writelines file handling mechanisms.

➢ **Advanced**

14	Write a program in python to implement Salary printing file read operation. (Fileformat: Employee No, name, deptno, basic, DA, HRA, Conveyance) should perform below operations. a) Print Salary Slip for given Employee Number b) Print Employee List for Given Department Number
15	Write a program in python to implement Railway Reservation System using filehandling technique. System should perform below operations. a) Reserve a ticket for a passenger. b) List information all reservations done for today's trains.
16	Write a Python program to implement module.
17	Write a program which will implement decorators for functions and methods in python.
18	Write a program to read CSV file and generate output using HTML table.
19	Write a program to process CSV file using CSV module.
20	Desirable: Write a program to process JSON and XML data.
21	Create Web Database Application "Address Book" with options to a) add/ insert a record b) modify a record

	c) display a record d) delete a record
22	Create Web Database Application "Event Registration" with options to a) Event Registration b) Cancel Registration c) display a record

Part II: Advanced Topic: Data Analysis

1	Perform following operations on a CSV file a. Create a data frame from csv file, dictionary, List of tuples b. Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data frame d. Finding maximum and minimum values e. Displaying statistical information f. Performing queries g. Data Analysis using groupby()

Part III: Advanced Topic: Data cleaning

1	Handling dirty data / missing data

Part IV: Advanced Topic: Python for Data Visualization
Library: pylab, matplotlib, seaborn

1	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object oriented paradigm is compulsory.)
2	Using a data file, draw a. Bar Graph b. Histogram c. Pie Chart d. Line Chart
3	Perform following operations on a CSV file a. Create a data frame from csv file, dictionary, List of tuples b. Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data frame d. Finding maximum and minimum values e. Displaying statistical information f. Performing queries g. Handling missing data

Part V: Advanced Python Programming: Regular Expressions

1	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both. (Note: Use of object orientedparadigm is compulsory.)
	a) Recognize following strings bit, but, bat, hit, hat or hut
	b) Match any pair of words separated by a single space, that is, first and last names.
	c) Match any word and single letter separated by a comma and single space, as in last name, first initial.
	d) Match simple Web domain names that begin with www and end with a ".com" suffix;for example, www.yahoo.com. Extra Credit: If your regex also supports other high-level domain names, such as .edu, .net, etc. (for example: www.foothill.edu).
	e) Match a street address according to your local format (keep your regex general enough to match any number of street words, including the type designation). For example, American street addresses use the format: 1180 Bordeaux Drive. Make your regex flexible enough to support multi-word street names such as: 3120 De la Cruz Boulevard.
2	Create utility script to process telephone numbers such that
	a. Area codes (the first set of three-digits and the accompanying hyphen) are optional, that is, your regex should match both 800-555-1212 as well as just 555-1212.
	b. Either parenthesized or hyphenated area codes are supported, not to mention optional; make your regex match 800-555-1212, 555-1212, and also (800) 555-1212.
3	Chapter End Practical List of Wesley J Chun, Core Python Applications Programming, 3rd Edition. Pearson

CO- PO Mapping:

Semester 2	**Course Name: Programming in Python**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	2	-	-	-	-
CO2	3	2	3	3	-	-	-	-
CO3	3	2	3	3	-	-	-	-
CO4	3	2	2	2	-	-	-	-
CO5	3	2	3	2	-	-	-	-
CO6	3	-	2	1	-	-	-	1

*\Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC02094061

Course / Subject Name: Indian Knowledge System

w. e. f. Academic Year:	2024-25
Semester:	2
Category of the Course:	Indian Knowledge System(IKS)

Prerequisite: Motivation to learn and explore great Indian heritage

Rationale:	INDIAN civilization has permanently attached great value to knowledge. In Bhagavad Gita, Shree Krishna says – "**न हि ज्ञानेन सदृशं पवित्रमिह विद्यते।**" - Indeed, there is nothing purifying here comparable to Knowledge. The Indian Knowledge System talks about two types of knowledge (**विध्या**), one **अपराविध्या** (worldly knowledge) and **पराविध्या/ब्रह्मविध्या** (knowledge which liberate). However, it is a common perception that the body of knowledge represented in our Indian Scriptures only relates to some ritual practices. On the contrary, there is an amazingly large body of intellectual texts, the world's largest collection of manuscripts, and its attested tradition of texts, thinkers, and schools in so many domains of knowledge. It is an exciting and fascinating aspect of knowledge in India that it prevails in diverse ways and is expressed at varied levels. In many areas such as Medicine, Mathematics, Science and Technology, Psychology, Philosophy, Agriculture, Grammar, Language, Dance, Music, and Astrology, to name just a few, there is wide and extensive knowledge both at the level of the classical texts and the folk traditions. They are often referred to as "Shastra" and "Lok Parampara" respectively. This course aims to introduce this great tradition of knowledge to BCA students. This module will introduce scientific and technological aspects of the great Indian Knowledge System.

Course Outcome:

After completion of the course, the student will able to:

No	Course Outcomes	RBT Level
01	Explain about various texts and traditions of the Indian Knowledge System.	U
02	Explain the scientific aspects of Sanskrit and its Grammar.	U
03	Explain Indian contribution to mathematics, science, technology, philosophy, health, and psychology.	U

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
2	0	0	2	70	30	0	0	100

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to IKS**	**2**	**5**
2.	**The Vedic Corpus**	**2**	**5**
3.	**Indian Philosophical System**	**3**	**10**
4.	**Scientific Foundation in Linguistics**	**3**	**10**
5.	**Indian Number System and Units of Measurement**	**3**	**10**
6.	**Knowledge: Framework and Classification**	**3**	**10**
7	**Indian Mathematics: Vedic Mathematics and other ancient mathematical work**	**8**	**40**
8	**Astronomy**	**2**	**5**
9	**Health, Wellness, and Psychology**	**2**	**5**
	Total	**28**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
50	**50**	**-**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Introduction to Indian Knowledge System: Concepts and Applications by B. Mahadevan, Vinayak Rajat Bhat, and Nagendra Pavana R.N.; Publisher: Prentice Hall India

2. Set of five books published by School of IKS (**धरोहर**),

CO- PO Mapping:

Semester 2	**Indian Knowledge System**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	-	-	-	-	-	-	-	1
CO2	1	-	-	-	1	-	-	1
CO3	-	-	-	-	1	-	-	1

Note: PO5 and PO7 are attained through group project on relevant topic.

GTU *Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code : MC03094011

Course / Subject Name : Design and Analysis of Algorithms

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Core Courses

Prerequisite:	Data Structure, C Programming Language.
Rationale:	The course intends to introduce the concepts of artificial intelligence and its applications. The course provides the opportunities for the learner to learn the artificial intelligence based basic methods and algorithms. This course also covers the knowledge representation and reasoning that help the learners to develop the intelligent systems

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Analyze the time and space complexity of algorithms and evaluate the growth of functions.	AN
02	Apply suitable algorithm design methodologies such as divide-and-conquer and randomized algorithms to solve problems.	AP
03	Design efficient solutions using the principles and elements of dynamic programming.	CR
04	Develop algorithms using greedy methods and implement solutions for minimum spanning trees and shortest paths.	AP
05	Solve complex computational problems using backtracking and branch-and-bound strategies.	AP

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction:** Role of Algorithms in Computing, Algorithms, Algorithms as a technology, Insertion sort, Analyzing algorithms, Designing algorithms,	5	10

	Growth of Functions: Asymptotic notation, Standard notations and common functions.		
2.	**Algorithms using Divide-and-Conquer, Probabilistic Analysis and Randomized Algorithms:** The maximum-subarray problem, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrences The hiring problem, Indicator random variables, Randomized algorithms.	9	20
3.	**Dynamic Programming:** Rod cutting, Matrix-chain multiplication, Elements of dynamic programming, Longest common subsequence, Optimal binary search trees.	11	25
4.	**Greedy Algorithms Minimum Spanning Trees and Single-Source Shortest Paths:** Example—Knapsack Problem, Job Sequencing with Deadlines, An activity-selection problem, Elements of the greedy strategy Growing a minimum spanning tree, The algorithms of Kruskal and Prim, The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra's algorithm	10	25
5.	**Backtracking, Branch and Bound Algorithms:** BFS, DFS, 8-Queen problem, M-Coloring problem, Hamiltonian circuits; **Branch-and-Bound algorithms:** Examples: Shortest path; 16-Puzzle and 8-Puzzle, 0/1 Knapsack problem, Traveling salesman problem; Limitations of Branch-and-Bound	10	20
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	20	30	20	10	10

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, Introduction to Algorithms, Third Edition, MIT Press (2009).
2. Parag H Dave, Himanshu B Dave, Design and Analysis of Algorithms Pearson (2014) Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran Fundamentals of Computer Algorithms Universities Press (2008)
3. S. Baase, Computer Algorithms: Introduction to Design and Analysis, Pearson (2002)

4. Aho, Hopcroft, and Ullman, The Design and Analysis of Computer Algorithms Addison Wesley
5. Kleinberg Algorithm Design, Pearson (2013)

(b) Open source software and website:

1. https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/
2. https://onlinecourses.nptel.ac.in/noc19_cs47/preview
3. https://www.w3schools.com/dsa/

Suggested Practical List, if any:

List of Mandatory Lab Exercises (Write Algorithms and Implement in C/C++ Language) for the following problems, students are expected to write one or more (as the case may be) algorithms along with the complexity of these algorithms, and implement them in C/C++ Language.

1. Find square root of a number. Can we use Divide & Conquer approach for this problem?
2. Determine smallest divisor of an integer.
3. For a given value of n, generate prime numbers <= n (more than one algorithms are possible)
4. Find X^n. Iterative and recursive algorithms are possible.
5. Find Factorial of n. Iterative and recursive algorithms are possible.
6. Generate Fibonacci series up to n terms Iterative and recursive algorithms are possible.
7. Program for finding maximum and minimum number using Divide and conquer.
8. Implement Recursive Binary search and Linear search and determine the time taken to search an element.
9. Sort a given sequence of numbers using (a) Bubble Sort, and (b) Merge Sort
10. Generate permutations of given n numbers. Iterative and recursive algorithms are possible.
11. Implement Strassen's algorithm for matrix multiplication
12. Implement knapsack using greedy algorithm.
13. Implement Matrix multiplication using Dynamic Programming algorithm.
14. Implement Rod-cutting problem using Dynamic Programming algorithm.
15. Implement LCS using Dynamic Programming algorithm.
16. Breadth First Search (BFS) in a binary tree.
17. Depth First Search (DFS) in a binary tree.
18. Prim's algorithm to find minimum cost tree (shortest path in a tree).
19. Kruskal's algorithm to find minimum cost tree (shortest path in a tree).
20. Implement Bellman-Ford Single Source Shortest Path Algorithm

Suggested Activities for Students, if any:

- ICT enabled Classroom teaching
- Case study
- Practical / live assignment
- Interactive class room discussions

CO- PO Mapping:

Semester 3	**Design and Analysis of Algorithms**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	-	-	-	-	-
CO2	3	3	2	-	-	-	-	-
CO3	3	3	3	-	-	-	-	-
CO4	3	3	3	-	-	-	-	-
CO5	3	3	3	-	-	-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094021

Course / Subject Name : Software Engineering

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Core Courses

Prerequisite:	Systems & Object Oriented Design Methodologies, Programming experience of higher level languages.
Rationale:	The course intends to introduce the concepts of software engineering and software development. The course provides the opportunities for the learner to learn the the fundamentals, software processes and agile development process. This course also covers the requirements engineering and its use in software modeling and design. This course equips the learner with the knowledge of software testing, reliability engineering, project and quality management.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Describe the concepts of software engineering, software process model and agile development.	UN
02	Discuss requirement engineering concepts and their use in system modeling.	UN
03	Discuss architectural design, object oriented design and related implementation.	UN
04	Describe various software testing and reliability engineering aspects.	UN
05	Discuss importance of project management and quality management.	UN

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
4	0	0	4	70	30	0	0	100

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction:** Professional Software development, Software engineering ethics, Case studies.	12	20

	Software processes: Software process models, Process activities, Coping with change, Process Improvement. **Agile software development:** Agile methods, Agile development techniques, Agile project management, Scaling agile methods.		
2.	**Requirements engineering:** Functional and nonfunctional requirements, Requirements engineering processes, Requirement elicitation, Requirement specification, Requirement validation, Requirements change. **System modeling:** Context models, Interaction models, Structural models, Behavioral models, Model driven architecture.	12	20
3.	**Architectural design:** Architectural design decisions, Architectural views, Architectural patterns, Application architecture. **Design and Implementation:** Object oriented design using the UML, Design patterns, Implementation issues, Open source development.	12	20
4.	**Software testing:** Development testing, Test-driven development, Release testing, User testing. **Reliability engineering:** Availability and reliability, Reliability requirements, Fault-tolerant architectures, Programming for reliability, Reliability measurement.	12	20
5.	**Project management:** Risk management, Managing people, Teamwork. **Quality management:** Software quality, Software standards, Review and inspections, Quality management and agile development, Software measurement.	12	20
	Total	**60**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	50	40	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

Text book:

- Sommerville , Software Engineering , 10th edition, Pearson

Reference Books:

- Roger S. Pressman, Software Engineering – A Practitioner's Approach, McGraw Hill Publications
- Chandramouli Subramanian, Saikat Dutt, Chandramouli Seetharaman,B G Geetha, Software Engineering, Pearson
- Waman S. Jawadekar, Software Engineering– Principles and Practices, TMGH Publication
- Blaha, Rumbaugh, Object Oriented Modeling and Design with UML, 2e, Pearson
- Pankaj Jalote, Software Engineering –A Precise Approach, Wiley India
- Behhforoz & Frederick Hudson, Software Engineering Fundamentals, OXFORD
- Rajib Mall, Fundamentals of software Engineering, Prentice Hall of India.

(b) Open source software and website:

- Faculty can suggest any online course from NPTEL, EdX, Coursera, Udemy , Agile and Scrum platforms (Based on availability of the course at the time of teaching learning as course availability remains changing.)

Various Web Based SE Tools

- Software:-Rational Rose, Microsoft Visio, Enterprise resource planning
- Project Management Tools
- SCM Tools
- SQA Tools
- Analysis and Design Tools
- User Interface Development Tools
- Object-Oriented Software Engineering Tools
- Testing Tools

Suggested Activities for Students, if any:

- ICT enabled Classroom teaching
- Case study
- Assignments
- Interactive class room discussions

CO- PO Mapping:

Semester 3	**Course Name: Software Engineering**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	1	2	-	-	2	1	-
CO2	3	3	3	1	-	-	-	-
CO3	3	2	3	1	-	-	-	-
CO4	3	2	3	1	-	-	-	-
CO5	2	-	1	-	-	3	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: PG

Course / Subject Code: MC03094031

Course / Subject Name : Mobile Application Development

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Core Courses

Prerequisite:	Knowledge of the Core Java Programming, database concepts
Rationale:	Smartphones and mobile applications have grown in popularity in recent years. This trend is expected to continue, resulting in an increased demand for professionals who can create mobile applications. Android mobile application development is a relevant and important topic for computer applications students because it allows them to apply their programming skills and knowledge to create real-world applications. The course provides students with hands-on experience in developing mobile applications using Android. This is essential for students to be able to apply the concepts they have learned in a real-world setting. This course will help students to build core competencies in mobile application development with relevant skills and knowledge, practical experience, and career opportunities in a growing and dynamic industry.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Describe various concepts of mobile programming that make it unique from programming for other platforms.	UN
02	Create, test and debug Android application by setting up Android development.	CR
03	Demonstrate methods in storing, sharing and retrieving data in Android applications.	AP
04	Use rapid prototyping techniques to design and develop sophisticated mobile interfaces.	CR
05	Create interactive applications in android using databases with multiple activities including audio, video and notifications and deploy them in marketplace.	CR

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Basics of ANDROID** Introduction to ANDROID: ANDROID SDK Features, Introduction to Development Features Developing for ANDROID, developing for mobile and embedded devices, ANDROID development tools Creating Applications using ANDROID Basics of an ANDROID application, introduction to manifest, externalizing resources, application lifecycle, ANDROID activities	**6**	**10%**
2.	**Use Interface in Android** Building user interfaces Introduction to layouts, introduction to fragments, creating new views, introduction to adapters Intents and broadcast receivers Introduction to intents, creating intents and broadcast receivers Using Internet resources Downloading and parsing internet resources, using the download manager, using internet services.	**9**	**25%**
3.	**Working with Data in Android** Files, saving state and preferences Creating, saving and retrieving share preferences, including static files as resources, working with the file system **Database and content providers** Introducing ANDROID databases, content values and cursors, working with SQLite databases, creating content providers, using content providers, native ANDROID content providers. **Working in background** Introducing services, using background threads, using alarms.	**9**	**25%**
4.	**Enhancing user experience** Introduction and addition of action bar, Creating and using menus and Action bars action menus, introducing dialogs, let us make toast, Introducing notifications, **Advanced User experience:** Designing for every Screen size and density, Introducing Android Text-to speech, Using speech recognition, Controlling Device vibration, Working with Animations, Enhancing your views, Advanced Drawable resources, Copy, paste and the clipboard.	**9**	**15%**
5.	**Advanced Android Audio, video and using the camera:** Playing audio and video, manipulating raw audio, using camera to take pictures, recording video, adding media to media store **Telephony and SMS:** Hardware support for telephony, using telephony, introducing SMS and MMS	**9**	**15%**
6.	**Publishing your App:** Monetizing, promoting and distributing the	**3**	**10%**

	applications, Signing and publishing applications, distributing applications, introduction to monetizing applications		
	Total	**45**	**100**
7.	**Web Services (Only for implementation in lab)** **Introduction, Web services Types, Consuming Web Services, Performance Case Study:** Simple Android Web service, How to call Java web services in Android, Android JSON or XML web services, Authenticate users using web service	-	-

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	30	30	10	10	10

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Lauren Darcey and Shane Conder,"Android Wireless Application Development", Pearson Education, 2nded.(2011)
2. Mark L Murphy,"Beginning Android",Wiley India Pvt Ltd (2009)
3. Sayed Y Hashimi and Satya Komatineni, "ProAndroid",Wiley India Pvt Ltd (2009)

(b) Open source software and website:

1. https://developer.android.com/index.html
2. Android Developer Fundamental Course by Google and Advance Android Developer Course by Google (https://developer.android.com/courses)

Suggested Project List, if any:

S. No.	Practical Outcomes (PrOs)	Approx. Hrs.
1	Set-up of Android development environment, managing AVD and understanding its various components.	02
2	Understanding of Various Components available in Android Application	02
3	Develop a "Hello World" Application in Android and understand the structure of an Android Application	02

4	Develop Android Application to demonstrate methods of Activity Life Cycle.	02
5	Design Android Activities using LinearLayout, RelativeLayout, GridView, FrameLayout, and ConstraintLayout	06
6	Design various Activities using different Layouts and available Widgets(TextView, EditText, Button, RadioButton, CheckBox, ImageButton, ToggleButton, TimePicker, DatePicker, ProgressBar, ImageView) to make the user-friendly GUI.	04
7	Develop code to demonstrate different ways of Handling different events (onClick, onLongClick etc.) over Button, EditText etc. to perform action in Android application at run-time.	04
8	Develop code to demonstrate Event handling of CheckBox and RadioButton selection.	02
9	Develop code to navigate between different activities and pass the data from one activity to other activity using Intent.	02
10	Develop an android application to store data locally using Shared Preferences and access-modify in different activities.	02
11	Develop the code to implement the ListView and the Spinner views, perform add, update, remove items operations and implement the item selection event handling over ListView and Spinner for appropriate example.	04
12	Develop the code to manage Permission using Manifest file and run time from Activity, and toggle state of WiFi and Bluetooth.	02
13	Develop android applications to demonstrate user interaction with the application using Options Menu, Context Menu and Popup Menu.	04
14	Develop Android Applications to demonstrate different Alert Dialogs and the Custom Dialog.	04
15	Develop Android Application for local database connectivity and performing basic database operations (select, insert, update, delete) using SQLiteDatabase and SQLiteOpenHelper Classes	04
17	Develop an Android Application to demonstrate the use of RecyclerView and CardView for displaying list of items with multiple information	04
18	Develop a simple application to display “Hello <Application Name>” using Kotlin	02
19	Develop an android application using Kotlin having a Button “Click” and upon clicking on that Button a Toast message “Button Clicked” should be displayed on screen through Toast Message	02
20	Publish an Android Application on Play Store	02

Note: More Practical Exercises can be designed and offered by the respective course teacher to develop the industry relevant skills/outcomes to match the COs. The above table is only a suggestive list.

CO- PO Mapping:

Semester 3	Course Name : Mobile Application Development							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	3	1	2	1	2
CO2	2	3	2	2	1	2	-	2
CO3	1	3	2	2	2	2	-	2
CO4	2	1	2	3	2	1	-	2
CO5	2	2	1	2	1	2	1	3

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094041

Course / Subject Name: Human Computer Interface

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-1

Prerequisite:	Basic knowledge of software development principles, programming fundamentals, and operating systems. Familiarity with user-level interaction on computer applications and exposure to GUI-based tools will be beneficial.
Rationale:	Human-Computer Interface (HCI) is a critical domain that bridges technology and human behavior. As computing systems become more integrated into everyday life, the design and evaluation of user-friendly, accessible, and intuitive interfaces become essential. This course aims to equip students with a foundational understanding of how users interact with technology and how to design systems that are efficient, effective, and satisfying to use. By incorporating principles from cognitive science, design, and usability engineering, students will be better prepared to create technology solutions that meet the diverse needs of users. The course also introduces emerging interaction paradigms such as voice, gesture, and virtual interfaces, preparing students for future interface innovations. **Pedagogy:** The course on Human-Computer Interface will follow an experiential and learner-centered pedagogy. It integrates interactive lectures with real-time case discussions, hands-on lab sessions, and iterative design practices. Emphasis will be placed on user-centered design thinking, encouraging students to empathize with users and iterate solutions based on usability feedback. Open-source tools will be employed for wireframing, prototyping, accessibility testing, and usability evaluation to build practical skills. Collaborative projects, peer reviews, and heuristic evaluations will nurture teamwork, creativity, and analytical thinking. The course also incorporates reflective learning through continuous feedback and demonstration-based assessments.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Describe the foundations of Human-Computer Interaction and its principles	UN
2	Analyze the user-centered design process and interaction paradigms	AN

3	Apply usability principles to design intuitive user interfaces	AP
4	Evaluate interface designs through heuristic evaluation and usability testing	EL
5	Create basic prototypes using HCI tools and techniques	CR

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	Introduction to HCI: Definition, human factors, history, HCI paradigms	6	13
2	Human Capabilities: Perception, cognition, memory, errors	7	15
3	Interaction Design Process: UCD, task analysis, scenarios, prototyping	10	22
4	Interface Design Principles: Guidelines, heuristics, visual design, accessibility	10	22
5	Usability Engineering and HCI in Emerging Tech: Usability testing, VR, AR, voice UI	12	28
	Total Hours:	**45**	**100%**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	U Level	A Level	N Level	E Level	C Level
5	20	20	20	20	15

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Open Source HCI Tools

1. Pencil Project
Use: GUI prototyping and wireframing
Platform: Windows, macOS, Linux
Website: https://pencil.evolus.vn
Use in Lab: Design simple wireframes for desktop or mobile apps

2. Inkscape
Use: Vector-based graphic design, UI component design
Platform: Cross-platform
Website: https://inkscape.org
Use in Lab: Create high-fidelity interface elements, icons, layouts

3. GazeRecorder (Free version)
Use: Eye-tracking for usability analysis (limited free use)
Platform: Windows
Website: https://www.gazerecorder.com
Use in Lab: Understand visual attention on screen layouts (optional with webcam eye-tracking)

4. NVDA (NonVisual Desktop Access)
Use: Screen reader for accessibility testing
Platform: Windows
Website: https://www.nvaccess.org
Use in Lab: Test if interfaces are usable by visually impaired users

5. Quant-UX
Use: UX design, interactive prototyping, and analytics
Platform: Web-based
Website: https://quant-ux.com
Use in Lab: Build and test interactive prototypes with user session tracking

6. Arch Accessibility Tools (Linux)
Use: Evaluate interfaces for WCAG compliance and assistive tech
Tools Include: Orca (screen reader), Accerciser (accessibility inspection)
Use in Lab: Simulate and test accessible application features

7. OpenViBE
Use: Brain-Computer Interface (BCI) design and simulation
Platform: Windows, Linux

Website: http://openvibe.inria.fr
Use in Lab: Explore advanced HCI like BCI for research projects

8. ClickHeat
Use: Open-source heatmap tracking tool
Platform: PHP-based, Web
Website: https://www.labsmedia.com/clickheat/index.html

Use in Lab: Track user interactions on web interfaces (as part of usability testing)

Reference Books:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale – Human-Computer Interaction, Pearson
2. Ben Shneiderman et al. – Designing the User Interface, Pearson
3. Donald Norman – The Design of Everyday Things, Basic Books
4. Jenny Preece – Interaction Design: Beyond Human-Computer Interaction, Wiley
5. Jeff Johnson – Designing with the Mind in Mind, Morgan Kaufmann
6. R. Jacob – Brain-Computer Interaction: A New Frontier, Springer
7. Steve Krug – Don't Make Me Think, New Riders

Suggested Course Practical List: if any

To perform followings Lab work:

Sr. No.	Title of Practical	Tool(s) used	Outcome Mapped
1	Introduction to wireframing and prototyping concepts	Pencil Project	CO1, CO3
2	Create low-fidelity wireframes for a mobile or desktop application	Pencil Project, Inkscape	CO3
3	Design UI components and layout using vector design principles	Inkscape	CO3
4	Create an interactive prototype with basic navigation	Quant-UX	CO3, CO5
5	Conduct heuristic evaluation on a peer group's prototype	Manual + Nielsen's Heuristics	CO2, CO4
6	Perform usability testing with task-based scenarios and record findings	Quant-UX	CO4
7	Analyze visual attention using webcam-based eye tracking (optional setup)	GazeRecorder (Free version)	CO4, CO5

8	Test the accessibility of a simple web form using screen readers	NVDA	CO4
9	Simulate a non-visual interface experience for visually impaired users	NVDA	CO4
10	Evaluate accessibility properties using Linux tools (Orca, Accerciser)	GNOME Accessibility Tools	CO4
11	Record and analyze click heatmaps for a simple HTML page	ClickHeat	CO4
12	Introduction to Brain-Computer Interface simulation and visualization	OpenViBE	CO5
13	Design a voice or gesture-based interaction scenario (concept + mock-up)	Manual/Tool-based	CO2, CO5
14	Final Project: Design and prototype a small multi-page interface with usability review	Tools as needed	CO3, CO4, CO5
15	Final Project Demonstration and Reflection	-	All COs

CO- PO Mapping:

Semester	**Course Name :**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	1	2	-	-	-	2
CO2	2	3	2	2	2	-	-	2
CO3	2	2	3	3	2	1	-	3
CO4	2	3	2	3	1	1	1	3
CO5	1	2	3	3	3	2	1	3

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094051

Course / Subject Name: Artificial Intelligence

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-1

Prerequisite:	Basic programming concepts, data structures, discrete mathematics, fundamentals of algorithms and complexity, probability theory
Rationale:	This course provides foundational knowledge in artificial intelligence. It is focusing on logical reasoning, problem-solving techniques, and uncertainty modeling essential for designing intelligent systems. The course equips students with practical skills in logic programming using PROLOG, heuristic search methods, and probabilistic reasoning, enabling them to develop effective AI solutions for real-world problems.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Apply formal logic systems to represent and reason with knowledge.	AP
2	Analyse the inherent limitations of classical logic in AI	AN
3	Develop logic program using PROLOG	AP
4	Implement search strategies and game-playing algorithms for effective problem-solving in AI	AP
5	Apply probabilistic reasoning and Bayesian networks to handle uncertainty	AP

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	**Introduction and Logics:** Basics of artificial intelligence, the history of AI, agents, knowledge-based systems. Propositional Logic: Syntax, Semantics, Proof Systems, Resolution, Horn Clauses, Computability and Complexity, Applications and Limitations First-order Predicate Logic: Logic Syntax, Semantics, Quantifiers and Normal Forms, Basic Examples of First-order Logic Statements	11	25
2	**Limitations of Logic:** The Search Space Problem, Decidability and Incompleteness, The Flying Penguin, Modeling Uncertainty	5	10
3	**Logic Programming with PROLOG:** PROLOG Systems and Implementations, Simple Examples, Execution Control and Procedural Elements, Lists, Self-modifying Programs, A Planning Example, Constraint Logic Programming	5	10
4	**Search, Games and Problem Solving:** Introduction, Uninformed Search, Breadth-First Search, Depth-First Search, Iterative Deepening, Comparison, Cycle Check, Heuristic Search, Greedy Search, A★-Search, Route Planning with the A★ Search Algorithm, IDA★-Search, Empirical Comparison of the Search Algorithms, Summary, Games with Opponents, Minimax Search, Alpha-Beta-Pruning, Non-deterministic Games, Heuristic Evaluation Functions, Learning of Heuristics, State of the Art.	12	30
5	**Reasoning with Uncertainty:** Reasoning with Uncertainty, Computing with Probabilities, Conditional Probability, The Principle of Maximum Entropy, An Inference Rule for Probabilities, Maximum Entropy Without Explicit Constraints, Conditional Probability Versus Material Implication, MaxEnt-Systems, The Tweety Example, Reasoning with Bayesian Networks, Independent Variables, Graphical Representation of Knowledge as a Bayesian Network, Conditional Independence, Practical Application, Software for Bayesian Networks, Development of Bayesian Networks	12	25
	Total Hours:	**45**	**100%**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	30	45	15	0	0

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Textbook:

1. Introduction to Artificial Intelligence, By: Wolfgang Ertel, 2nd Edition, Springer

References Books:

1. Artificial Intelligence A Modern Approach, By: Stuart Russell and Peter Norving, 3rd Edition, Pearson Publication
2. First Course In Artificial Intelligence, By Deepak Khemani, Mc Graw Hill
3. Artificial Intelligence, By Elaine Rich and Kevin Knight, 2nd Edition, Mc Graw Hill
4. Artificial Intelligence: Making a System Intelligent, By: Dr. Nilakshi Jain, As per AICTE, Wiley Publication
5. Introduction to Artificial Intelligence, By: Eugene Charniak and Drew McDermott, Addison-Wesley Series in Computer Science
6. Artificial Intelligence: An Engineering Approach, By: Robert J. Schalkoff, Mc Graw Hill,
7. Programming in Prolog: Using the ISO Standard, By: William F. Clocksin, Christopher S. Mellish, 5th Edition, Springer

List of Useful websites / MOOCs

1. Learners are advised to opt for NPTEL and SWAYAM courses that are relevant to this course

Suggested Course Practical List:

To perform followings Lab work:

S/W Required: GNU Prolog, SWI-Prolog, MATLAB, Python, Code block C/C++

1. Define family relationships (parent, child, sibling, grandparent) using facts and rules.
2. Identify all ancestors of a person using recursive rules.
3. Create a knowledge base of animals and classify them as birds, mammals, etc., and query their properties.
4. Implement a simple expert system to identify diseases based on symptoms.
5. Define arithmetic operations using recursive logic (e.g., addition, factorial).

6. Represent and solve a basic logic puzzle (e.g., who owns the zebra?).
7. Create a list-based program to find the maximum of a list.
8. Use lists to reverse elements and query the reversed list.
9. Implement a simple database of books and authors and query based on genre or author.
10. Model and reason about traffic lights (e.g., when is it safe to cross?).
11. Represent family tree data and infer cousin or uncle relationships.
12. Define a rule-based planning system to plan a simple sequence of tasks (e.g., preparing tea).
13. Simulate an AI agent's decision-making using facts and rules.
14. Model a simple library management system and allow checking book availability.
15. Create a timetable conflict checker for classrooms using constraints.
16. Use lists and recursion to calculate the length of a list.
17. Model a set of rules for granting university admission based on criteria.
18. Represent propositional logic formulae and test satisfiability (manually).
19. Simulate a basic flight route planner using simple search rules.
20. Define and simulate the game of Tic-Tac-Toe with basic moves.
21. Write a knowledge base for different animals and identify which can fly (Flying Penguin example).
22. Define and implement rules for sorting a list of numbers using recursion.
23. Represent and evaluate first-order logic statements (e.g., all humans are mortal, Socrates is a human).
24. Simulate a basic constraint logic programming example like Sudoku (3x3).
25. Model a simple chatbot using pattern matching and rules for basic conversation.
26. Implement Breadth-First Search to find a path between two cities in a connected graph.
27. Model a maze and solve it using Depth-First Search in PROLOG.
28. Simulate a simple two-player game (e.g., Nim game) with Minimax strategy.
29. Develop an A search algorithm for finding the shortest route between locations with given heuristics.

City	Heuristic (h) to Goal
A	10
B	6
C	4
D	7
E	2
Goal	0

30. Implement Alpha-Beta Pruning in a simple game tree (like a reduced Tic-Tac-Toe tree).
31. Simulate uncertain knowledge by assigning confidence levels to rules and use rule selection based on thresholds.
32. Model the Tweety example using exceptions and rules to handle uncertainty (e.g., birds generally fly unless they are penguins).

33. Implement a rule-based system where multiple causes can lead to a symptom, and the program ranks possible causes.
34. Design a simplified Bayesian-like reasoning system using rule weights or scoring system (simulated in PROLOG).
35. Build a basic expert system for weather prediction using rule-based inference with uncertain facts (e.g., "if clouds and humidity are high, rain is likely").

CO- PO Mapping:

Semester : 3	**Course Name : Artificial Intelligence**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	2	-	-	-	-	-
CO2	3	2	2	1	-	-	-	-
CO3	2	1	3	3	-	-	-	-
CO4	3	3	3	2	-	-	-	-
CO5	3	2	2	2	-	-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094061

Course / Subject Name: Augmented Reality/Virtual Reality

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-1

Prerequisite:	Basic knowledge of computer graphics, programming fundamentals, and linear algebra.
Rationale:	This course introduces the foundational concepts and tools used in Augmented Reality (AR) and Virtual Reality (VR). With the increasing use of immersive technologies in various industries, this course provides students with essential theoretical and practical knowledge using open-source platforms. Students will develop interactive AR/VR experiences and understand their application in education, healthcare, gaming, manufacturing, and more. **Pedagogy:** The course will be delivered through a blend of lectures, demonstrations, and project-based learning. Emphasis will be on experiential learning using open-source AR/VR platforms such as AR.js, A-Frame, and Unity with open plugins. Students will engage in hands-on lab sessions and mini-projects to build immersive applications. Regular assessments through quizzes, demos, and peer evaluations will ensure concept clarity and skill development. Collaborative activities and real-world case studies will enhance problem-solving and design thinking abilities.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Explain the core concepts and technological components of AR and VR systems.	UN
2	Compare AR and VR platforms and frameworks based on features and usability.	AN
3	Develop basic AR applications using open-source tools like AR.js or Unity with open-source plugins.	AP
4	Create VR scenes and interactive content using tools like A-Frame or OpenXR-based engines.	CR
5	Evaluate challenges and ethical aspects in AR/VR system design and deployment.	EL

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	Introduction to AR and VR: Evolution, definitions, applications; differences between AR, VR, MR, XR; hardware requirements; sensors; displays.	9	20%
2	AR Systems: Marker-based and markerless tracking; AR development tools overview (AR.js, Vuforia with Unity OpenXR plugin); Mobile AR concepts.	9	20%
3	VR Systems: VR devices, tracking and controllers; concepts of locomotion, field of view, immersion, and presence; using A-Frame for VR development.	9	20%
4	Interaction and Design in AR/VR: UI/UX for immersive systems; gestures and interaction models; spatial sound; design guidelines and ergonomics.	9	20%
5	Ethics, Challenges, and Case Studies: Ethical issues, privacy and data security, motion sickness, accessibility; real-world applications and case studies in healthcare, education, etc.	9	20%
	Total Hours:	**45**	**100%**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	U Level	A Level	N Level	E Level	C Level
10%	20%	30%	20%	10%	10%

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Reference Books:

1. Chetankumar G. Shetty, "Augmented Reality & Virtual Reality", Notion Press, 2022 (Open educational content)
2. Jason Jerald, "The VR Book: Human-Centered Design for Virtual Reality", ACM Books, 2015
3. Alan B. Craig, "Understanding Augmented Reality: Concepts and Applications", Morgan Kaufmann
4. Tony Parisi, "Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile", O'Reilly Media
5. Joseph Hocking, "Unity in Action", Manning Publications (with focus on open-source compatible plugins)

Suggested Course Practical List: if any

To perform followings Lab work:

1	Set up development environment for AR/VR using open-source platforms.
2	Develop a basic marker-based AR application using AR.js.
3	Implement a markerless AR app for Android using Unity and AR Foundation.
4	Create a 3D virtual environment using A-Frame.
5	Add interaction and navigation to the virtual scene.
6	Integrate audio and spatial sound in VR scenes.
7	Experiment with gesture recognition using webcam-based input.
8	Conduct a usability test on designed AR/VR content.
9	Develop a mini-project using either AR or VR frameworks.

List of Laboratory/ Active Learning Assignment: If any

- Weekly coding assignments and tool configuration.
- Individual practical demos.
- Team-based AR/VR experience design project.

Platforms for Hands-on Learning

- AR.js Official Tutorial (https://github.com/AR-js-org/AR.js)
- A-Frame Documentation (https://aframe.io/docs/)
- Mozilla Hubs (Free VR spaces) (https://hubs.mozilla.com/)
- Unity Learn – Free AR/VR Projects (https://learn.unity.com/)
- 8thWall WebAR Demo Playground (https://www.8thwall.com/demos)

- Case Studies Collection: XR for Everyone – Unity Learn (Free) - https://learn.unity.com/project/xr-for-everyone
- Glitch.com – Open A-Frame Templates (https://glitch.com/@aframe)

CO- PO Mapping:

Semester	**Course Name :** Augmented Reality/Virtual Reality							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	-	-	-	-	1	-
CO2	2	3	-	2	-	-	1	-
CO3	2	2	3	3	2	-	-	-
CO4	2	3	3	3	3	-	-	-
CO5	2	2	2	1	-	2	2	2

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094071

Course / Subject Name: Containerization

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-1

Prerequisite:	Basic knowledge of Linux commands, networking, and software development life cycle. Familiarity with virtualization concepts is helpful but not mandatory.
Rationale:	Containerization is a key technology in modern software engineering and DevOps. It allows applications to be packaged with their dependencies and run consistently across environments. This course introduces the foundational concepts and practical tools for working with containers and orchestration platforms like Docker and Kubernetes. Students will gain hands-on skills to build, deploy, and manage containerized applications, making them industry-ready for cloud-native development environments. **Course Pedagogy:** The course will follow a hands-on, application-driven pedagogy that blends theory with practical exposure. Each concept will be introduced through real-world problems, followed by live demonstrations using open-source tools like Docker and Kubernetes. Students will engage in interactive lab sessions to reinforce their understanding and build containerized applications. Assignments and mini-projects will foster collaboration and critical thinking. Online labs and simulation platforms will be used for experiential learning and to encourage self-paced exploration beyond the classroom.

Course Outcome:

After Completion of the Course, students will be able to:

No	Course Outcomes	RBT Level
01	Understand containerization principles and architecture.	UN
02	Deploy and manage containers using Docker.	AP
03	Use Docker Compose to define and run multi-container applications.	AP
04	Explain and implement container orchestration using Kubernetes.	AN
05	Apply containerization to real-world software development and DevOps scenarios.	CR

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Introduction to Containerization: Virtualization vs. Containerization, Container lifecycle, Benefits, Use cases	9	20
2.	Docker Basics: Docker architecture, Images, Containers, Volumes, Networks, Docker CLI	9	20
3.	Docker Compose: Writing docker-compose files, Linking services, Environment variables, Networking	9	20
4.	Container Orchestration: Kubernetes architecture, Pods, Services, Deployments, ConfigMaps, Helm	9	20
5.	CI/CD and DevOps Integration: Using containers in build pipelines, Security and best practices, Monitoring	9	20
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	20	40	10	10	10

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Reference Books:

1. Nigel Poulton, *Docker Deep Dive*, Independently Published, 2023
2. Kelsey Hightower, Brendan Burns, Joe Beda, *Kubernetes: Up and Running*, O'Reilly Media
3. Pethuru Raj, Jeeva S. Chelladhurai, Vinod Singh, *Docker in Practice*, Manning Publications
4. Elton Stoneman, *Learn Docker – Fundamentals of Docker 19.x*, Packt Publishing

(b) Online Learning Resources and Opensource software and website:

- Docker – https://www.docker.com
- Docker Compose – https://docs.docker.com/compose/
- Kubernetes – https://kubernetes.io
- Minikube – https://minikube.sigs.k8s.io/docs/

Suggested List of Practical

1. Install and configure Docker on Linux/Windows.
2. Create, run, stop, and manage Docker containers using CLI.
3. Build Docker images from custom Dockerfiles.
4. Use Docker Volumes and Networks in applications.
5. Write Docker Compose files and deploy multi-container setups.
6. Setup Minikube for Kubernetes cluster simulation.
7. Deploy containers on Kubernetes using kubectl.
8. Create and manage Kubernetes Services and Deployments.
9. Integrate Docker containers in a simple CI/CD pipeline.
10. Monitor container resources and apply security best practices.

CO- PO Mapping:

Semester 3	**Course Name: Containerization**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	-	-	-	-	-
CO2	3	3	3	2	-	-	-	-
CO3	3	2	3	2	1	-	-	-
CO4	3	3	3	3	2	-	-	-
CO5	3	3	3	3	2	1	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094081

Course / Subject Name: Big Data Tools

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-1

Prerequisite:	Basic understanding of databases, data structures, and programming concepts. Familiarity with SQL and any programming language (preferably Python or Java) is recommended.
Rationale:	• To build a strong foundation in big data concepts, architectures, and tools used for handling large-scale data. • To provide hands-on experience with leading big data technologies such as Hadoop, NoSQL, MongoDB, Hive, Pig, and Spark. • To understand data storage, processing, and analysis in distributed computing environments. • To develop practical skills for managing, querying, and analyzing structured and unstructured data. • To enable students to apply big data tools and techniques in solving real-world problems and to pursue careers in fields leveraging large-scale data technologies.

Course Outcome:

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

No	Course Outcomes	RBT Level
01	Discuss the core concepts of Big Data, including its characteristics, challenges, and evolving ecosystem.	UN
02	Apply the concepts of NoSQL to develop the non relational databases using MongoDB	AP
03	Implement the various concepts of Hadoop, Hadoop ecosystem, Hadoop Components, HDFS and Map Reduce in managing large-scale data environments.	CR
04	Design and analyze data processing workflows using HIVE and Pig	AP
05	Explain the basic concepts of SPARK	UN

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA(I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Big Data** Types of Digital Data: Classification of Data (Structured, Semistructured and Unstructured), Characteristics of Data, Evolution of Big Data, Definition of Big Data, Challenges of Big Data, Characteristics of Big Data (Volume, Velocity, Variety), Other characteristics of Big Data which are not Definitional Traits of Big Data, Why Big Data? Are we Information Consumer or Producer? Traditional BI vs Big Data, Typical Data Warehouse Environment, Typical Hadoop Environment, what is Changing in Realms of Big Data? Terminologies used in Big Data Environments	5	15%
2.	**Introduction to NoSQL and Hadoop** **NoSQL:** Introduction: Where is it used? What is it? Types of NoSQL databases, Why NoSQL?, Advantages of NoSQL, Use of NoSQL in Industry, SQL vs NoSQL, NewSQL **Hadoop:** Introduction, Distributed Computing Challenges, History of Hadoop, Overview of Hadoop and Hadoop Ecosystems, Features and key advantages of Hadoop, Versions of Hadoop, Hadoop distributions, RDBMS versus Hadoop, Hadoop vs SQL, Integrated Hadoop Systems offered by leading market vendors, Cloud based Hadoop solutions, HDFS, Processing data with Hadoop, Managing Resources and applications with Hadoop YARN, Interacting with Hadoop Ecosystem	13	25%
3.	**Introduction to MongoDB and Map Reduce** **MongoDB:** Introduction: What is MongoDB? Why Mongo DB? (using JSON, Creating or generating a unique key, Support for Dynamic Queries, Storing Binary Data, Replication, Shading, Updating information in place), Terms used in RDBMS and Mongo DB, Data types in Mongo DB, MongoDB Query Language **Map Reduce:** Data Flow, Map, Shuffle, Sort, Reduce, Hadoop Streaming, mrjob, Installation, word count in mrjob, Executing mrjob	12	25%
4.	**Introduction to HIVE and Pig** **HIVE:** Introduction: What is HIVE? HIVE Architecture, HIVE data Types, HIVE File Formats, HIVE Query Language, RCFile implementation, SerDe, User-Defined Functions (UDF). **Pig:** Introduction: What is Pig? The anatomy of Pig, Pig on Hadoop, Pig philosophy, Use Case for Pig- ETL Processing, Pig Latin overview, Data types in Pig, Running Pig, Execution modes of Pig, HDFS commands, Relational operators, Eval function, Complex Data Types, Piggy Bank,	10	25%

	User-defined Functions, Parameter substitution, Diagnostic Operator, Word Count Example using Pig, when to use and not use Pig?, Pig at Yahoo, Pig vs HIVE.		
5.	Overview of SPARK Introduction to Data Analysis with Spark, Downloading Spark and Getting Started, Programming with RDDs.	5	10%
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	25	30	25	5	5

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Seema Acharya, Subhashini Chellappan, " Big Data and Analytics", Wiley India Pvt. Ltd.,2015
2. Matei Zaharia, Patrick Wendell, Andy Konwinski, Holden Karau, "Learning Spark", O'Reilly Media,2015
3. Shashank Tiwari, "Professional NoSQL", Wiley India Pvt. Ltd.,2011
4. Kyle Banker,Peter Bakkum,Shaun Verch,Douglas Garrett,Tim Hawkins,"MongoDB in Action", DreamTech Press, 2nd Edition ,2016
5. Chris Eaton,Paul Zikopoulos,Tom Deutsch,George Lapis,Dirk Deroos,"Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data", Mcgraw Hill Education (India)Pvt.Ltd.,2012
6. Tom White,"Hadoop: The Definitive Guide",O'Reilly Media,4th Edition,2015
7. Vignesh Prajapati,"Big Data Analytics With R and Hadoop", Packt Pub Ltd ,2013

(b) Web Resources:

1. http://www.bigdatauniversity.com
2. http://www.mongodb.com
3. http://hadoop.apache.org/

(b) Open-source software and website:

1. Development Environment & IDEs
   - Apache Zeppelin – https://zeppelin.apache.org/
   - Jupyter Notebook – https://jupyter.org/
   - Visual Studio Code – https://code.visualstudio.com/
2. Big Data Frameworks
   - Apache Hadoop – https://hadoop.apache.org/
   - Apache Spark – https://spark.apache.org/
   - Apache Hive – https://hive.apache.org/
   - Apache Pig – https://pig.apache.org/

3. NoSQL & Databases
   - MongoDB (Community Edition) – https://www.mongodb.com/try/download/community
   - Apache HBase – https://hbase.apache.org/
   - CouchDB – https://couchdb.apache.org/
   - Cassandra – https://cassandra.apache.org/
4. Data Processing & Querying Tools
   - Apache Flink – https://flink.apache.org/
   - Apache Drill – https://drill.apache.org/
5. Cluster & Resource Management
   - Apache YARN - https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html
   - Apache Mesos – https://mesos.apache.org/
   - Kubernetes – https://kubernetes.io/
6. Cloud & Distributed Systems
   - Google Cloud Big Data Tools – https://cloud.google.com/products/big-data
   - AWS Big Data Solutions – https://aws.amazon.com/big-data/
   - Microsoft Azure HDInsight – https://azure.microsoft.com/en-in/products/hdinsight/

SUGGESTED PROJECT LIST:

Note: Hadoop programs can be implemented using either Java or Python

MongoDB

1. **Create a Student Master Database**
   Create a database named StudentDB and a collection called Student containing documents with the following fields: StudentRollNo, StudentName, Grade, Hobbies.
   Perform the following operations:
   a) Insert at least 5 student records.
   b) Update the grade of a student.
   c) Delete a student record.
   d) Retrieve all students whose grade is above 'B'.
2. **Employee Management System**
   Create a database CompanyDB with a collection Employees. Each document should have: EmpID, EmpName, Department, JoiningDate, Salary.
   Perform queries to:
   a) Find employees from a specific department.
   b) Sort employees by salary in descending order.
   c) Find employees who joined after 2022.
3. **Use of Aggregation Framework**
   Use any existing collection (e.g., Employees from above) and perform the following tasks:
   a) $group to calculate total salary by department.
   b) $match to filter records by condition.
   c) $project to include/exclude specific fields.
   d) $sort based on a numeric field.
4. **Online Store Inventory System**

Create a collection Products with fields: ProductID, ProductName, Category, Price, StockAvailable.
Tasks:

a) Insert 10 product documents.
b) Find products in the 'Electronics' category.
c) Update stock for a particular product.
d) Delete out-of-stock products.

5. **Library Database**
Create a Library collection with fields: BookID, Title, Author, Genre, AvailableCopies.
Perform:
a) Insert multiple book records.
b) Find all books by a specific author.
c) Increment the number of copies for a book when new stock arrives.
d) Use $in operator to find books in 'Fiction' or 'Science' genres.
6. **Customer Feedback System**
Create a collection Feedback with: FeedbackID, CustomerName, Rating, Comments, Date.
Perform the following:
a) Insert 5 feedback entries.
b) Find feedback with rating >= 4.
c) Use projection to retrieve only CustomerName and Rating.
d) Count total feedback entries.
7. **Working with Date and Time**
Create a collection Events with fields: EventID, EventName, EventDate, Location.
Performed the following tasks:
a) Insert 5 events with different dates.
b) Retrieve upcoming events (EventDate > current date).
c) Format the date using MongoDB operators.
d) Sort events based on date.

Hadoop/HDFS

8. **Install and Set Up Hadoop Environment**
a) Install Java Development Kit (JDK)
b) Download and extract the Hadoop package
c) Configure environment variables for Java and Hadoop
d) Verify the installation of Java and Hadoop using version commands
9. **HDFS – Basic File Operations**
Perform the following tasks using Hadoop HDFS commands:
a) Create a file in the local file system and add some sample content.
b) Create a directory in HDFS.
c) Upload the local file to the HDFS directory using different commands (-put, -copyFromLocal).
d) List the contents of the HDFS directory (-ls, -ls -R).
e) Display the contents of the file in HDFS using the -cat command.
f) Copy the file within HDFS using the -cp command.
g) Download the file back to the local system using -get or -copyToLocal.

h) Delete the file and directory from HDFS using -rm -r.

10. Word Count using mrjob (Python-based MapReduce)

Using mrjob library in Python:

a) Create a Python script to count the frequency of words in a text file
b) Run the job locally and on Hadoop (if configured)

HIVE/PIG

11. Create a Hive table and load data

a) Create a Hive table with sample fields.
b) Load data from a local or HDFS file.
c) Display all records using a SELECT query.

12. Perform basic SQL operations in Hive

a) Use SELECT, WHERE, GROUP BY, and ORDER BY clauses.
b) Run queries to filter and sort data.

13. Load data in Pig and view structure

a) Load a data file using Pig Latin.
b) Use DUMP and DESCRIBE to view content and schema.

14. Filter and group data using Pig

a) Apply FILTER and GROUP operations.
b) Count grouped results using FOREACH and COUNT.

15. Use functions and expressions in Pig

a) Use eval functions like TOKENIZE, UPPER, or SUBSTRING.
b) Apply transformations on loaded data.
c) View results using DUMP.

16. Perform word count using Pig Latin

a) Load a text file.
b) Split lines into words.
c) Count frequency of each word and store results.

CO- PO Mapping:

Semester 3	**Big Data Tools**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	-	-	-
CO2	3	2	2	3	-	-	-	-
CO3	3	2	2	3	-	-	-	-
CO4	3	3	3	3	-	-	-	-
CO5	3	3	-	3	-	-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094091

Course / Subject Name: Business Analytics

w. e. f. Academic Year:	2025 – 26
Semester:	3
Category of the Course:	Elective Group-1

Prerequisite:	Basic knowledge of data handling and spreadsheet tools (e.g., MS Excel). A fundamental understanding of databases, data types, and data structures. Introductory-level statistics and logical reasoning are beneficial but not essential.
Rationale:	• To build a strong foundation in business intelligence and data visualization concepts using industry-leading tools. • To gain practical, hands-on experience with Power BI and Tableau for real-world data analysis. • To develop skills in data modeling, dashboard creation, and interactive reporting for informed decision-making. • To enhance analytical thinking and the ability to communicate insights effectively through visual storytelling. • To prepare students for career opportunities in data analytics, business intelligence, and related technology-driven roles.

Course Outcome:

After Completion of the Course, the Student will be able to:

No	Course Outcomes	RBT Level
01	Explain the core concepts of business intelligence and the significance of data visualization in decision-making using Power BI and Tableau.	UN
02	Demonstrate the ability to connect to, clean, and model data from various sources using Power BI and Tableau.	UN
03	Create dynamic dashboards and visual reports that effectively communicate business insights.	CR
04	Analyze real-world datasets to identify patterns, trends, and actionable business insights.	AN
05	Evaluate and compare the visualization strategies and analytical capabilities of Power BI and Tableau for solving business problems.	EV

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA(I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Business Intelligence and Power BI** Understanding Business Intelligence (BI) concepts, Importance of data visualization in decision-making, Overview of Power BI, Installing and setting up Power BI, Navigating the Power BI interface.	9	20%
2.	**Data Handling and Modeling in Power BI** Importing and connecting to various data sources, Data cleaning and transformation using Power Query, Creating and managing data models, Implementing calculated columns and measures using DAX.	9	20%
3.	**Advanced Analytics and Visualization in Power BI** Designing interactive reports and dashboards, Utilizing slicers, filters, and drill-down features, Incorporating time intelligence functions, Publishing and sharing reports via Power BI Service.	9	20%
4.	**Introduction to Tableau and Data Visualization** Overview of Tableau and its interface, Connecting to various data sources, Creating basic visualizations (bar charts, line graphs, maps), Building interactive dashboards.	9	20%
5.	**Advanced Features and Analytics in Tableau** Implementing calculated fields and table calculations, Using parameters and level of detail (LOD) expressions, Applying forecasting and trend analysis, Sharing and publishing dashboards.	9	20%
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	20	30	20	10	10

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

[Unit 1 to 3]

1. Clark, Dan. *Beginning Microsoft Power BI: A Practical Guide to Self-Service Data Analytics*. 2nd ed., Apress, 2020.
2. Deckler, Greg. *Learn Power BI: A Comprehensive, Step-by-Step Guide for Beginners to Learn Real-World Business Intelligence*. 2nd ed., Packt Publishing, 2022.
3. Ferrari, Alberto, and Marco Russo. *Introducing Microsoft Power BI*. Microsoft Press, 2016. Download a free eBook from https://download.microsoft.com/download/0/8/1/0816F8D1-D1A5-4F60-9AF5-BC91E18D6D64/Microsoft_Press_ebook_Introducing_Power_BI_PDF_mobile.pdf

[Unit 4 to 5]

4. Milligan, Joshua N. *Learning Tableau 2022: Create Effective Data Visualizations, Build Interactive Visual Analytics, and Improve Your Data Storytelling Capabilities*. 5th ed., Packt Publishing, 2022.
5. Milligan, Joshua N. *Learning Tableau 2025: Leverage Tableau's Newest Features to Revolutionize Your Data Storytelling with AI-Enhanced Insights*. 6th ed., Packt Publishing, 2025.
6. Sleeper, Ryan. *Practical Tableau: 100 Tips, Tutorials, and Strategies from a Tableau Zen Master*. O'Reilly Media, 2018.

(b) Free Learning Resources:

Power BI

- Microsoft Power BI course for free: https://learn.microsoft.com/en-us/training/courses/pl-300t00
- Training modules and courses: https://learn.microsoft.com/en-us/training/browse/?products=power-bi
- Online learning resources: https://learn.microsoft.com/en-gb/power-bi/

Tableau

- E-Learning resources: https://www.tableau.com/learn/training/elearning
- Free training videos: https://www.tableau.com/learn/training
- On-demand and live webinars to stay up to date: https://www.tableau.com/events/webinars

Suggested Project List:

A data file link is provided for each practical, but students are free to use any relevant data sets to perform the tasks.

Power BI

1. **Import and Clean Sales Data**
   Import a CSV file into Power BI and perform data cleaning operations such as removing null values, renaming columns, and changing data types.
   Dataset: Sales & Returns Sample
2. **Create Relationships and Data Model**
   Establish relationships between multiple tables (e.g., Customers, Orders, Products) and create a data model suitable for analysis.
   Dataset: AdventureWorksDW Excel Sample
3. **Build Calculated Columns and Measures**
   Use DAX to create calculated columns (e.g., Profit = Sales - Cost) and measures (e.g., Total Sales, Average Discount).
   Dataset: Financial Sample Workbook
4. **Design Interactive Dashboards**
   Create an interactive dashboard with slicers, filters, and multiple visualizations to analyze sales performance.
   Dataset: Retail Analysis Sample
5. **Implement Time Intelligence Functions**
   Apply DAX time intelligence functions to calculate Year-to-Date (YTD), Month-to-Date (MTD), and previous year comparisons.
   Dataset: Sales & Returns Sample
6. **Create KPI Indicators and Cards**
   Develop Key Performance Indicators (KPIs) and card visuals to display critical metrics like revenue and customer count.
   Dataset: Financial Sample Workbook
7. **Visualize Data with Maps**
   Use map visuals to represent sales data geographically, highlighting regions with the highest and lowest sales.
   Dataset: Retail Analysis Sample
8. **Publish Reports to Power BI Service**
   Publish a Power BI report to the Power BI Service, configure workspace settings, and share the report with stakeholders.
   Dataset: Any of the above samples.

Tableau

Use the superstore dataset from Tableau Sample Data for all the following demonstrations:

9. **Connect to Data and Create Basic Visualizations**
   Connect Tableau to a dataset and create basic visualizations such as bar charts and line graphs.

10. **Build Interactive Dashboards with Filters and Actions**
    Use the same dataset to combine sheets into dashboards with filter actions and interactivity.
11. **Create Calculated Fields and Use Parameters**
    Create calculated fields and parameters using the dataset to enable dynamic data filtering.
12. **Develop Geo Maps and Color Code by Sales**
    Plot sales on geographic maps with color coding based on the dataset.
13. **Create a Story to Present Business Insights**
    Use the dataset to build a story that guides viewers through business insights sequentially.
14. **Publish Tableau Dashboard to Tableau Public or Server**
    Publish the dashboard created from this dataset to Tableau Public or Server with sharing configurations.

CO- PO Mapping:

Semester 3	**Business Analytics**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	3	-	-	-	2
CO2	3	3	2	3	-	-	-	-
CO3	2	3	3	3	2	-	1	-
CO4	3	3	3	3	-	-	1	2
CO5	3	3	2	3	-	-	1	2

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094101

Course / Subject Name : Information and Network Security

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-1

Prerequisite:	Fundamentals of computer network **,** Mathematical concepts: Random numbers, Number theory, finite fields
Rationale:	The use of the Internet for various purpose including social, business, communication and other day to day activities has been in common place. The information exchanged through Internet plays vital role for their owners and the security of such information/data is of prime importance. Knowing the concepts, principles and mechanisms for providing security to the information/data is very important for the students of MCA. The subject covers various important topics concern to information security like symmetric and asymmetric cryptography, hashing, message and user authentication, digital signatures, key distribution and overview of the malware technologies. The subject also covers the applications of all of these in real life applications.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Explain concepts of cryptography.	UN
02	Implement symmetric key cryptography algorithms.	AP
03	Implement asymmetric key cryptography algorithms.	AP
04	Apply concepts of hashing, digital signature and MAC.	AP
05	Analyze security vulnerabilities and breaches of network and use appropriate approach and tools to mitigate it.	AN

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1	Introduction – Security attacks, services & security mechanisms, CIA (Confidentiality, Integrity, Availability), AAA (Authentication, Authorization, Accounting), Symmetric Cipher Model, Cryptography, Cryptanalysis and Attacks; Substitution and Transposition techniques	6	15
2	Stream ciphers and block ciphers, Block Cipher structure, Data Encryption standard (DES) with example, AES with structure, its transformation functions, key expansion, example	6	15
	Multiple encryption and triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback mode, Output Feedback mode, Counter mode	5	10
3	Public Key Cryptosystems with Applications, Requirements and Cryptanalysis, RSA algorithm, its computational aspects and security, Diffie-Hillman Key Exchange algorithm, Man-in-Middle attack	6	15
4	Cryptographic Hash Functions, their applications, Simple hash functions, its requirements and security, Secure Hash Algorithm (SHA)	5	10
	Message Authentication Codes , Digital Signature, its properties, NIST digital Signature algorithm	6	15
5	Concepts of DNS,DHCP, Public & Private IP address, NAT, CIDR, DMZ, Firewall, IDS, IPS, WHOIS, NSLOOKUP, TRACERT, ICMP Ping	4	5
	port scanning using Nmap, Traffic Capturing & Packet Analysis using wireshark	2	5
	Security at Application layer-HTTPS, Security at transport layer-TLS,SSL	3	5
	Security at internet layer-IPSec Security at Network interface layer-Network access control(MAC filtering)	2	5
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	50	30	10	0	0

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Cryptography And Network Security, Principles And Practice , William Stallings, Pearson
2. Information Security Principles and Practice By Mark Stamp, Willy India Edition
3. Cryptography & Network Security, Forouzan, Mukhopadhyay, McGrawHill
4. Cryptography and Network Security Atul Kahate, TMH
5. Cryptography and Security, C K Shyamala, N Harini, T R Padmanabhan, Wiley-India
6. Information Systems Security, Godbole, Wiley-India
7. Information Security Principles and Practice, Deven Shah, Wiley-India
8. Security in Computing by Pfleeger and Pfleeger, PHI
9. Build Your Own Security Lab : A Field Guide for network testing, Michael Gregg, Wiley India

(b) Open source software and website:

1. https://www.wireshark.org/
2. https://nmap.org/
3. https://www.whois.com/
4. https://www.vlab.co.in/

Suggested Practical List, if any:

1.	Implement Caesar cipher.
2.	Implement Brute Force attack on Caesar cipher.
3.	Implement Mono-alphabetic cipher.
4.	Implement Play-fair cipher.
5.	Implement Hill cipher.
6.	Perform DES encryption technique with cryptool.
7.	Write a program to implement RSA algorithm.
8.	Write a program to implement Diffi-Hellmen Key exchange Method.
9.	Perform digital signature algorithm with cryptool.
10.	Perform various types of port scanning using Nmap/Zenmap.
11.	Study and demonstrate use of ping, tracert, nslookup, whois
12.	Demonstrate use of Wireshark to capture and analyze the network traffic.

Suggested Activities for Students, if any:

- Study of security architecture of your institute's network.
- Study recent case studies related to this course.
- Visit national as well as state level cyber crime portal.
- Study and discuss different types of cyber attacks.

- Study of tools to launch different cyber attacks and preventive ways for the same.

CO- PO Mapping:

Semester-3	**Information and Network Security**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	3	-	-	1	-
CO2	3	3	2	3	-	-	-	-
CO3	3	3	3	3	-	-	-	-
CO4	3	3	1	2	-	-	-	-
CO5	3	3	2	3	-	-	-	3

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094111

Course / Subject Name: DevOps Engineering

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-2

Prerequisite:	Basic knowledge of software development life cycle (SDLC), familiarity with command-line tools, and hands-on experience with any programming language (e.g., Python, Java, or JavaScript). Understanding of source code version control systems (e.g., Git) is desirable but not mandatory.
Rationale:	In today's fast-paced digital world, software products must be developed, tested, and deployed rapidly while maintaining quality and reliability. DevOps bridges the gap between software development and IT operations, enabling faster delivery cycles, improved collaboration, and continuous improvement. This course introduces students to the foundational principles and tools of DevOps, equipping them with practical skills in version control, CI/CD pipelines, automation, containerization, and orchestration. Students learn to work with industry-standard tools such as Git, Jenkins, Docker, Kubernetes, and Ansible—preparing them to contribute effectively to modern agile and cloud-native development environments. The course emphasizes hands-on experience and real-world scenarios, making learners job-ready and responsive to evolving industry practices. It aligns with industry demands for professionals who can ensure faster time-to-market, better operational efficiency, and collaborative development practices. **Pedagogy:** The course follows a blended learning approach, combining interactive lectures with extensive hands-on lab sessions to reinforce practical skills. Concepts are delivered through case studies, tool demonstrations, and real-world scenarios to contextualize DevOps practices. Students engage in collaborative projects, peer learning, and problem-solving activities using tools like Git, Jenkins, Docker, and Kubernetes. Capstone tasks and mini-projects ensure experiential learning. The course also incorporates self-paced learning resources and continuous assessment through quizzes, assignments, and lab work to track progress and enhance retention.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Describe the fundamentals and need of DevOps and Agile methodologies.	UN
2	Apply version control using Git and GitHub effectively.	AP
3	Demonstrate CI/CD pipeline automation using Jenkins.	AP
4	Use containerization technologies like Docker and orchestration using Kubernetes.	AP
5	Implement DevOps tools and practices in real-world scenarios.	CR

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
2	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	Introduction to DevOps, Agile vs DevOps, DevOps lifecycle	2	6%
2	Source Code Management – Git, GitHub basics, branching strategies	2	6%
3	GitHub Actions and collaboration workflows	2	6%
4	Introduction to Continuous Integration (CI)	2	7%
5	Jenkins: Installation, configuration, and plugins	2	7%
6	Automating builds and tests with Jenkins	2	7%
7	Continuous Delivery & Continuous Deployment concepts	2	7%
8	Configuration Management: Ansible basics	2	7%
9	Containerization: Docker fundamentals, images, containers	2	7%
10	Docker Compose, DockerHub, Dockerfile	2	7%
11	Kubernetes basics: Pods, Services, ReplicaSets	2	7%

12	Kubernetes deployment and orchestration	2	7%
13	Monitoring and Logging (e.g., ELK, Prometheus, Grafana)	2	7%
14	DevOps Security and Best Practices	2	6%
15	DevOps Capstone Discussion and Project Integration	2	6%
	Total Hours:	**30**	**100%**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	40	30	-	-	20

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Reference Books:

1. "The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations" by Gene Kim, Patrick Debois, John Willis, and Jez Humble – IT Revolution Press.
2. Accelerate: The Science of Lean Software and DevOps" by Nicole Forsgren, Jez Humble, and Gene Kim – IT Revolution.
3. "Site Reliability Engineering: How Google Runs Production Systems" by Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy – O'Reilly Media.
4. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley – Addison-Wesley.
5. "Learning Continuous Integration with Jenkins" by Nikhil Pathania – Packt Publishing.
6. "Docker Deep Dive" by Nigel Poulton – Leanpub.
7. "Kubernetes Up & Running: Dive into the Future of Infrastructure" by Brendan Burns and Joe Beda – O'Reilly Media.
8. "Ansible for DevOps" by Jeff Geerling – Leanpub.

Online Courses and MOOCs:

1. DevOps on Coursera – Offered by IBM, AWS, or Google Cloud on topics like CI/CD, Jenkins, Docker, Kubernetes.
2. Linux Foundation's DevOps Tools and Practices – edX.org.
3. YouTube Channels:
    a. TechWorld with Nana

b. The Net Ninja
c. freeCodeCamp

Documentation and Tool Resources:

- Git Documentation
- Jenkins Documentation
- Docker Documentation
- Kubernetes Documentation
- Ansible Documentation

Suggested Course Practical List: if any

1. Setting up Git and GitHub account, basic Git operations
2. Creating and managing branches, merge conflicts in Git
3. Writing and using GitHub Actions for automation
4. Installing and setting up Jenkins
5. Automating a simple build pipeline in Jenkins
6. Integration of Jenkins with GitHub
7. Creating a simple Ansible playbook
8. Installing Docker, creating and running Docker containers
9. Writing a Dockerfile and building a custom image
10. Using Docker Compose to run multi-container applications
11. Kubernetes: Deploying a simple app using kubectl
12. Creating and managing Kubernetes pods and services
13. Setting up Prometheus and Grafana for basic monitoring
14. Implementing security practices in DevOps pipelines
15. Capstone: End-to-end CI/CD pipeline for a sample application

CO- PO Mapping:

Semester 3	**Course Name : DevOps engineering**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	2	1	2	1	-	-	2
CO2	2	3	2	3	2	1	-	2
CO3	2	2	3	3	2	2	-	2
CO4	2	2	3	3	2	2	1	3
CO5	3	3	3	3	3	3	2	3

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: PG

Course / Subject Code: MC03094121

Course / Subject Name : Machine Learning

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-2

Prerequisite:	Basic math (algebra, probability, and statistics), programming (preferably Python), and an understanding of data structures. Some knowledge of data analysis is helpful but not required.
Rationale:	• To develop a strong foundation in machine learning concepts, algorithms, and applications. • To gain hands-on experience with Python and machine learning tools for practical implementation. • To understand and apply supervised and unsupervised learning techniques, feature engineering, and neural networks. • To enhance problem-solving and analytical thinking skills for real-world machine learning applications. • To acquire the necessary knowledge and expertise for careers in data science, artificial intelligence, and research.

Course Outcome:

After Completion of the Course, the Student will able to:

No	Course Outcomes	RBT Level
01	Differentiate various learning approaches, and to interpret the concepts of supervised learning and unsupervised learning.	UN
02	Apply various techniques for data preprocessing & evaluating and improving performance of model	AP
03	Apply concepts of feature engineering, bayes' theorem and naive bayes' classifier	AP
04	Use different supervised learning algorithms on the given dataset	AP
05	Apply different clustering & association analysis algorithms on the dataset	AP

**Revised Bloom's Taxonomy (RBT): RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA(I)	ESE (V)	
2	-	2	3	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Machine Learning:** Human learning versus machine learning, types of machine learning, Applications of machine learning, tools for machine learning, Machine Learning Activities, Data structures for machine learning	4	16%
2.	**Preparing to Model, Feature Engineering:** Data Pre-processing Selecting a model, training a model, model Representation and interpretability Evaluating performance of a model, improving performance of a model.	6	21%
3.	**Modelling and Evaluation:** Introduction to feature engineering Feature transformation, feature subset selection, Concept learning through Bayes' theorem	6	21%
4.	**Supervised Learning:** Classification Model, k -Nearest Neighbour (kNN) algorithm, Decision tree algorithm Simple and multiple Regression algorithms, Assumptions in regression, Problems in regression, Accuracy improvement of linear regression.	7	21%
5.	**Unsupervised Learning:** Unsupervised Learning versus Supervised Learning, Applications of Unsupervised Learning, Clustering, Finding Pattern using Association Rules	7	21%
	Total	**30**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	30	60	0	0	0

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. **Machine Learning** – Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, *Pearson Education.* **(Main Book as per Syllabus Contents)**
2. **Machine Learning** – Tom M. Mitchell, *McGraw Hill*
3. **Machine Learning** – Anuradha Srinivasaraghavan, Vincy Joseph, *Wiley India*
4. **Machine Learning in Action** – Peter Harrington, *DreamTech*
5. **Real-World Machine Learning** – Henrik Brink, Joseph Richards, Mark Fetherolf, *DreamTech*

(c) Open-source software and website:

1. Development Environment & IDEs
   - Anaconda Distribution – https://www.anaconda.com/

- Jupyter Notebook – https://jupyter.org/
- VS Code (with Python Extension) – https://code.visualstudio.com/

2. Programming Languages
   - Python – https://www.python.org/
3. Open-Source Machine Learning Libraries
   - Scikit-Learn – https://scikit-learn.org/
   - Keras – https://keras.io/
   - PyTorch – https://pytorch.org/
4. Data Manipulation & Visualization Tools
   - NumPy – https://numpy.org/
   - Pandas – https://pandas.pydata.org/
   - Matplotlib – https://matplotlib.org/
   - Seaborn – https://seaborn.pydata.org/
5. Datasets & Data Sources
   - Kaggle Datasets – https://www.kaggle.com/datasets
   - UCI Machine Learning Repository – https://archive.ics.uci.edu/ml/index.php
   - Google Dataset Search – https://datasetsearch.research.google.com/

Suggested Project List:

The practical exercises should be performed in Python, preferably using a Python notebook.

1. Write a Python program to perform feature engineering on a Titanic Dataset (https://www.openml.org/d/42438).
   Perform the following tasks:
   a. Handling missing values
   b. Encoding categorical variables
   c. Scaling numerical features
   d. Creating new features
2. Write a Python program to perform Principal Component Analysis (PCA) on a given dataset. The objective is to reduce the dimensionality of the data while retaining the maximum variance. Use the dataset provided below and visualize the transformed features in a lower-dimensional space. Use wine dataset from UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/109/wine)
3. Write a Python program to implement a Decision Tree Classifier to predict whether a passenger survived the Titanic disaster based on passenger details. Evaluate the model using precision, recall, and F1-score. (Download dataset from https://www.openml.org/d/42438)
4. Implement a Python program that predicts house prices based on square footage using Linear Regression. Train the model on the given dataset and evaluate it using Mean Squared Error (MSE) and R^2 score. (Download dataset https://www.kaggle.com/datasets/yasserh/housing-prices-dataset/data)
5. Write a Python program to predict a car's fuel efficiency (MPG - miles per gallon) based on multiple factors such as horsepower, weight, and engine size using Multiple Linear Regression. Evaluate the model using R^2 score and RMSE. (Download dataset from https://archive.ics.uci.edu/dataset/9/auto+mpg)
6. Write a Python program to implement the k-Nearest Neighbors (kNN) algorithm to classify different species of iris flowers based on their petal and sepal measurements. Use different values

of k and evaluate the model using accuracy and a confusion matrix. (Download dataset from https://archive.ics.uci.edu/dataset/53/iris)

7. Write a Python program to implement the K-Means clustering algorithm to segment customers based on just two features: Annual Income and Spending Score. Use the K-Means algorithm to group customers into clusters and visualize the results with a simple scatter plot.
   Perform the following tasks:
   - Load the dataset and explore the data.
   - Standardize the features to ensure that both features contribute equally.
   - Apply K-Means Clustering to group the customers into 2 clusters.
   - Visualize the results using a scatter plot. Highlight the clusters with different colors.

Customer	Annual Income (k$)	Spending Score (1-100)
1	15	39
2	25	81
3	45	62
4	55	23
5	65	95
6	75	44
7	85	35
8	95	75
9	105	15
10	120	50

8. Write a Python program to implement Hierarchical Clustering (Agglomerative Clustering) to group social media users based on their interaction patterns. Use the following dataset consists of two features:
   - Number of Likes per Week
   - Number of Comments per Week

   Use Agglomerative Clustering to find natural groupings in the data and visualize the dendrogram to determine the optimal number of clusters.

User ID	Likes per Week	Comments per Week
1	5	2
2	20	8
3	30	12
4	40	20
5	60	30
6	70	35
7	80	40
8	90	50
9	100	55
10	110	65

9. Write a Python program to implement the Apriori Algorithm for market basket analysis. The objective is to identify frequent itemsets and generate association rules from the following dataset of customer transactions in a retail store.

Transaction ID	Items Purchased
1	Milk, Bread
2	Milk, Bread, Butter
3	Bread, Butter
4	Milk, Butter
5	Bread, Butter, Eggs
6	Milk, Bread, Eggs
7	Milk, Bread, Butter, Eggs
8	Bread, Eggs
9	Milk, Eggs
10	Milk, Butter, Eggs

Suggested Activities for Students (Optional):

1. Train multiple algorithms (e.g., Decision Tree, Random Forest, SVM) on the same dataset and compare accuracy.
2. Perform cross-validation and hyperparameter tuning to improve performance.
3. **Mini Project:** Fake News Detection - Apply Logistic Regression or SVM to classify real vs. fake news.
4. Deploy the model on Google Colab, Hugging Face Spaces, or Heroku.

CO- PO Mapping:

Semester 3	**Machine Learning**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	-	-	-
CO2	3	3	3	3	-	-	-	-
CO3	3	3	3	3	-	-	-	-
CO4	3	3	3	3	-	-	-	-
CO5	3	3	3	3	-	-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094131

Course / Subject Name: Cloud Computing

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-2

Prerequisite:	Operating System, Computer Networking, Distributed Computing, Computer Architecture
Rationale:	• This course aims students to understand the basic concepts of parallel and distributed computing and their role in cloud computing. • Students will realize the importance of cloud virtualization and relevant technologies available in the market. • Students will also learn about various cloud computing services, cloud platforms, cloud architecture, cloud service providers and cloud security. • This subject focuses on the many strategic and contemporary uses of cloud computing.

Course Outcome:

After Completion of the Course, Students will be able to:

No.	Course Outcomes	RBT Level*
1	Describe basic concepts and characteristics of distributed computing, Cloud Computing and Virtualization techniques	UN
2	Analyze the suitability of public, private, hybrid, and community cloud deployment models for different cloud service models (IaaS, PaaS, and SaaS).	AN
3	Describe the challenges of data security and privacy issues in the cloud environment.	UN
4	Explain the risk, consequences and estimate the cost for various Cloud Computing services.	UN
5	Gain exposure to Google App Engine and use the GAE launcher to launch the web applications.	AP

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
2	0	2	3	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	**Overview of Distributed Computing**: What is Computing? Traditional Utilities, Creation of the Internet, Computing Paradigm Trends, Computing Paradigm Evolution, Next Big Thing: The "Cloud", Cloud Computing: A New Paradigm, Differences and Similarities Among Different Types of Computing. **Introduction to Cloud Computing:** Cloud Computing: Definitions, Central Ideas Behind Cloud Computing, Properties and Characteristics of Cloud Computing, Benefits of Cloud Computing, Cloud Service and Deployment Models, Organizational Scenario of Cloud: Cloud Deployment Model, Cloud Architecture, Challenges with Cloud Computing, Cloud Supporting Services, Management and Administration of Cloud Services **Virtualization Techniques:** Virtualization Technology, Overview of X86 Virtualization, Types of Virtualization, Virtualization Products, Concept of VLAN and Benefits, Concept of SAN and Benefits, VM Migration, VM Consolidation and Management, Cloud Interoperability Standards	7	20
2	**Cloud Computing Services:** Infrastructure as Services (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Database as a Service (DBaaS) **Cloud Types and Models:(2)** Private Cloud, Community Cloud, Public Cloud, Hybrid Cloud **Application Architecture for Cloud:** Cloud Application Requirements, Recommendation for Cloud Application Architecture, Fundamental Requirements for Cloud Application Architecture, Relevance and use of Client-server Architecture for Cloud Applications, Addressing Cloud Application Performance and Scalability, Service Oriented Architecture (SOA) for Cloud Applications	7	25

3	**SLA with Cloud Service Providers:** The Concept of SLA, SLA Aspects and Requirements, Service Availability, Cloud Outages, Credit Calculation for SLA Breaches, Sample SLA for: Amazon S3, The Rackspace Cloud Server, Google Apps, HP Cloud Compute **Data Security in the Cloud:** Challenges with Cloud Data, Challenges with Data Security, Data Confidentiality and Encryption, Data Availability, Data Integrity, Cloud Data Management Interface, Cloud Storage Gateways (CSGs), Cloud Firewall, Virtual Firewall	6	25
4	**Risk, Consequences and Costs for Cloud Computing:** Introducing Risks in Cloud Computing, Risk Assessment and Management, Risk of Vendor lock-in, Risk of Loss of Control, Risk of Not Meeting Regulatory Compliance, Risk of Resource Scarcity or Poor Provisioning, Risk of Multi-Tenant Environment, Risk of Failure, Risk of Failure of Supply Chain, Risk of In adequate SLA, Risk of Malware and Internet Attack, Risk of Management of Cloud Resources, Risk of Network Outage, Risk in the Physical Infrastructure, Legal Risk due to Legislation, Risk with Software and Application Licensing, Security and Compliance Requirements in a Public Cloud, Calculating Total Cost of Ownership (TCO) for Cloud Computing, Direct and Indirect Cloud Costs, Costs Allocation in a Cloud, Chargeback Model for Allocation of Direct and Indirect Cost, Chargeback Methodology, Cost, Billable Items, Atomic Units, Pricing Model, Chargeback Tools and Solution, Maintaining Strategic Flexibility in a Cloud	6	15
5	**Using Google Web Services** Exploring Google Applications, Surveying the Google Application Portfolio, Exploring the Google Toolkit: The Google APIs, Working with the Google App Engine.	4	15
	Total Hours:	**30**	**100%**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	**40**	**40**	**10**	**0**	**0**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Textbook: (Unit-1 from Book-1, Unit-2 to 4 from Book-2 and Unit-5 from Book-3)

1. Cloud Computing: Fundamental Industry Approach and Trends, By: Rishabh Sharma, Wiley Publication, ISBN: 978-81-265-5306-8
2. Cloud Computing – Black Book, By: Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde and Dr. Deven Shah, Dreamtech Publication, ISBN: 978-93-5119-418-7
3. Cloud Computing - Bible, By: Barrie Sosinsky, Wiley Publishing Corporation, ISBN: 978-0-470-90356-8

Reference Books:

1. Mastering Cloud Computing, By: Rajkumar Buyya, Christian Vecchiola and S. ThamaraiSelvi, Elsevier, ISBN: 978-0-12-411454-8
2. Cloud Computing: Master the Concepts, Architecture and Applications with Real-world examples and Case studies, By: Kamal Kant Hiran, Ruchi Doshi, Dr. Temitayo Fagbola and Mehul Mahrishi, BPB Publication, ISBN: 978-93-88511-407
3. Cloud Computing: Principles and Paradigms, By: Rajkumar Buyya, James Broberg and Andrzej Goscinski, Wiley Publication, ISBN 978-0-470-88799-8
4. Cloud Computing: A Practical Approach, By: Anthony T Velte, Toby J. Velte, Robert Elsenpeter, Mc Graw Hill Publication, ISBN: 978-0-07-162695-8
5. Cloud Computing, By: Sandeep Bhowmik, Cambridge University Press, ISBN 978-1-316-63810-1
6. Programming Google App Engine, By: Dan Sanderson, 2nd Edition, O'Reilly, ISBN: 978-1-449-39826-2

List of Useful websites / MOOCs

1. https://www.tutorialspoint.com/cloud_computing/index.htm
2. https://www.geeksforgeeks.org/cloud-computing/
3. Learners are advised to opt for NPTEL and SWAYAM courses that are relevant to this course

Suggested Course Practical List:

1	Create an Account to Cloud Service Provider (AWS, AZURE, Google Cloud, etc.)
2	Create an Instance on Cloud
3	Provide Access Control and Permission to Users
4	Execute the Web Page on Cloud
5	Provide Security Mechanism to your instance.

6	Install Virtual-box / VMware Workstation with different flavors of Linux or windows OS on top of windows7 or 8.
7	Install a C compiler in the virtual machine created using virtual box and execute Simple Programs.
8	Install Google App Engine. Create hello world app and other simple web applications using python/java.
9	Use GAE launcher to launch the web applications.
10	Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
11	Find a procedure to transfer the files from one virtual machine to another virtual machine.
12	Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
13	Install Hadoop single node cluster and run simple applications like word count.
14	Creating Amazon EC2 instances with Microsoft Windows & Linux Server.
15	Launch a Web Server (IIS & Apache)
16	Working with Amazon Elastic Block Store (EBS)
17	Introduction to AWS Identity and Access Management (IAM)
18	Deploy a Web Application on AWS
19	CloudWatch Monitoring, Creating Dashboard, Setting up Alarms and Notifications.
20	Develop a hello world program web application and deploy it on the Google app engine.
21	Develop a web application which is customized version of the practical no 2, which relates the clock with Google accounts. Each user will get different view based on the preferences and the user's time zone.
22	Develop a simple Java web application for Google App Engine--a guestbook that lets users post messages to a public message board.
23	**List of projects:** (1) Online Book Store using Cloud Computing (2) University Campus Online Automation Using Cloud Computing (3) Student Information using Cloud Computing

CO- PO Mapping:

Semester : 3	**Course Name : Cloud Computing**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	1	1	1	-	-	-	-

CO2	2	2	-	1	-	-	-	-
CO3	2	2	-	-	-	-	-	-
CO4	3	1	2	1	-	2	-	-
CO5	3	2	2	3	-	2	1	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications
Level: Post Graduate
Course / Subject Code: MC03094141
Course / Subject Name: Software Testing

w. e. f. Academic Year:	2024-25
Semester:	3
Category of the Course:	Elective Group - 2

Prerequisite:	• Basic understanding of software development life cycle • Introductory knowledge of software testing • Familiarity with object-oriented programming concepts
Rationale:	Software testing is a crucial phase of the software development lifecycle. This course emphasizes advanced test design techniques with a strong foundation in graph-based, logic-based, and input domain modeling techniques. Learners are exposed to the theoretical and practical applications of automated testing strategies in various domains such as object-oriented and web-based applications, enabling them to deliver reliable and fault-tolerant systems. **Course Pedagogy:** The course will follow an open-source-first teaching model emphasizing hands-on learning. Students will use only open-source tools for test design, implementation, and evaluation. Case studies and coding assignments will be based on real-world problems and public repositories. Tool-based exploration and collaborative lab work will enhance understanding of test strategies.

Course Outcome:

After Completion of the Course, students will be able to:

No	Course Outcomes	RBT Level
01	Apply graph-based testing methods to ensure structural and data flow coverage.	AP
02	Analyze logic-based test criteria and apply them to specification models.	AN
03	Design effective test cases for object-oriented and web-based applications.	CR
04	Evaluate symbolic and concolic testing techniques for program analysis.	EL
05	Construct test cases using mutation, syntax-based and input partitioning methods.	CR

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
2	0	2	3	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Graph-based testing: Structural coverage (node, edge, path), Data flow coverage using open-source tools like GraphWalker.	9	20
2.	Graph coverage in source code, design and requirements. Logic-based testing using JaCoCo, CodeCover.	9	20
3.	FSM-based testing with GraphWalker, specification-based testing, input space partitioning with ACTS and PICT.	9	20
4.	Syntax-based testing using ANTLR and Grammarinator, mutation testing using PIT and MutPy.	9	20
5.	Symbolic testing using KLEE, concolic testing using CREST and Angr, application to OO and web applications using JUnit and Selenium.	9	20
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	20	25	15	15	15

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Reference Books:

1. Paul Ammann & Jeff Offutt, "Introduction to Software Testing", Cambridge University Press.
2. Aditya P. Mathur, "Foundations of Software Testing", Pearson Education.
3. Mauro Pezze & Michal Young, "Software Testing and Analysis: Process, Principles, and Techniques", Wiley.
4. Andreas Zeller, "Why Programs Fail: A Guide to Systematic Debugging", Elsevier.
5. Robert Binder, "Testing Object-Oriented Systems: Models, Patterns, and Tools", Addison-Wesley.

(b) Online Learning Resources and Opensource software and website:

- NPTEL Software Testing: https://nptel.ac.in/courses/106105150/
- GraphWalker: https://graphwalker.github.io/
- PIT (Java): https://pitest.org/
- MutPy (Python): https://mutpy.readthedocs.io/

- JaCoCo: https://www.eclemma.org/jacoco/
- KLEE: https://klee.github.io/
- ANTLR: https://www.antlr.org/
- Selenium: https://www.selenium.dev/
- Playwright: https://playwright.dev/
- ACTS (NIST): https://csrc.nist.gov/projects/automated-combinatorial-testing-for-software

Suggested List of Practical

1. Design graph-based test cases using GraphWalker.
2. Apply data flow test techniques and measure coverage using JaCoCo.
3. Derive logic-based test cases and validate with CodeCover.
4. Construct FSM models and generate tests using GraphWalker.
5. Model input domains using ACTS/PICT and generate test combinations.
6. Perform mutation testing using PIT (Java) and MutPy (Python).
7. Develop symbolic test cases using KLEE.
8. Implement concolic testing using CREST or Angr.
9. Write unit tests for OO programs using JUnit or PyTest.
10. Perform web application testing using Selenium or Playwright.

List of Active Learning Assignments:

- Evaluate and compare test coverage tools (e.g., JaCoCo vs. Coverage.py)
- Analyze test effectiveness in open-source projects.
- Mutation testing on student-developed code using PIT or MutPy.
- Syntax-based test case generation using ANTLR and Grammarinator.

CO- PO Mapping:

Semester 3	**Course Name : Containerization**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	-	-	-	-
CO2	2	3	2	3	-	-	-	-
CO3	2	2	3	3	-	-	-	-
CO4	2	2	2	3	-	-	-	-
CO5	2	2	3	2	-	-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094151

Course / Subject Name: Internet of Things

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Elective Group-2

Prerequisite:	Fundamentals of computer network, wireless sensor network, embedded systems, communication and internet technology, information security.
Rationale:	• Internet of Things plays an important role in connecting the things i.e. variety of devices through the Internet. • The IoT has emerged as a cutting-edge technology with applications in manufacturing, healthcare, agriculture, transport, mining, smart cities & many more • This subject covers the fundamentals of IoT with its architecture, protocols and applications. • It also covers the overview and programming of two widely used IoT platforms Raspberry Pi.

Course Outcome:

After Completion of the Course, Student will able to:

No.	Course Outcomes	RBT Level*
1	Describe basics concepts of IoT, things, smart connecting devices, IP & network layers, application protocols	UN
2	Apply different Sensors and Actuators in IoT Application as required	AP
3	Describe key IoT protocols (messaging, transport, and addressing)	UN
4	Develop basic applications using Raspberry Pi platforms.	AP
5	Discuss IoT security principles	UN

*RM: Remember, UN: Understand, AP: Apply, AN: Analyze, EL: Evaluate, CR: Create

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
2	0	2	3	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	Weightage (%)
1	**Introduction to Internet of Things (IoT):** Introduction and Definition of Internet of Things, Application Areas of IoT, Characteristics of IoT, Things in IoT, IoT Stack, Enabling Technologies, IoT Challenges, IoT Levels, Cyber Physical System and IoT, WSN and IoT	4	15
2	**Introduction to Sensors, Microcontrollers and Their Interfacing:** Introduction to Sensor Interfacing, Types of Sensors, Controlling Sensors through Webpages, Microcontrollers: A Quick Wallkthrough, ARM,	5	15
3	**Protocols for Internet of Things:** ***Messaging Protocols:*** Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP) ***Transport Protocols:*** Bluetooth Low Energy (BLE), Light Fidelity (Li-Fi), ***Addressing and Identification Protocols:*** Internet Protocol Version 4 (IPv4), Internet Protocol Version 6 (IPv6), Uniform Resource Identifier (URI)	5	15
4	**Raspberry Pi:** Raspberry Pi Architecture, Compatible Peripherals, Add-Ons and Accessories, Operating System for Raspberry Pi, Setting up Raspberry Pi, Initial Configuration for Raspberry Pi, Linux Based Software in Raspberry Pi, Application Development with Raspberry Pi-A Quick Walk Through **Application Building with IoT:** Various Application of IoT: Smart Perishable Tracking, Smart Healthcare, Smart Inflight Lavatory Maintenance, Water Quality Monitoring. Smart Warehouse Monitoring, Smart Retail, Smart Driver Assistance Systems Measure Collision Impact in an Accident	10	35
5	**IoT Security:** Cyber Security Vernacular, Anatomy of IoT Cyber Attacks, Physical and Hardware Security, Cryptograph, Software Defined Perimeter, Blockchains and Cryptocurrencies in IoT, Government Regulations and Intervention, IoT Security Best Practices	6	20
	Total Hours:	**30**	**100%**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	**40**	**50**	**0**	**0**	**0**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Textbook: (Unit-1 to 4 from Book-1 and Unit-5 from Book-2)

1. Internet of Things, By: Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, Wiley Publication, ISBN 978-81-265-7837-5
2. Internet of Things for Architects, By: Perry Lea, Packt Publication, ISBN 978-1-78847-059-9

Reference Books:

1. Designing the Internet of Things, By: Adrian McEwen, Hakim Cassimally, Wiley Publication, ISBN 978-1-118-43062-0
2. Internet of Things: Architecture and Design Principles, By: Dr. Raj Kamal, Mc Graw Hill Publication, ISBN-13: 978-93-5260-522-4
3. Internet of Things: Principles and Paradigms, By: Rajkumar Buyyaand and Amir Vahid Dastjerdi, Elsevier, ISBN: 978-0-12-805395-9

List of Useful websites / MOOCs

1. Learners are advised to opt for NPTEL and SWAYAM courses that are relevant to this course

Suggested Course Practical List:

To perform followings Lab work:

List of Experiments *using Raspberry Pi*

1	**Serial Monitor, LED, Servo Motor – Controlling**: (1) Controlling actuators through Serial Monitor. (2) Creating different led patterns and controlling them using push button switches. (3) Controlling servo motor with the help of joystick.
2	**Distance Measurement of an object:** (1) Calculate the distance to an object with the help of an ultrasonic sensor and display it on an LCD
3	**LDR Sensor, Alarm and temperature, humidity measurement:** (1) Controlling relay state based on ambient light levels using LDR sensor. (2) Basic Burglar alarm security system with the help of PIR sensor and buzzer. (3) Displaying humidity and temperature values on LCD

4	**Experiments using Raspberry Pi** (1) Controlling relay state based on input from IR sensors (2) Interfacing stepper motor with Raspberry-Pi (3) Advanced burglar alarm security system with the help of PIR sensor, buzzer and keypad. (Alarm gets disabled if correct keypad password is entered) (4) Automated LED light control based on input from PIR (to detect if people are present) and LDR (ambient light level)
5	**IOT Framework:** (1) Upload humidity and temperature data to Thing Speak, periodically logging ambient light level to Thing Speak (2) Controlling LEDs, relay and buzzer using Blynk app
6	**HTTP Based:** (1) Introduction to HTTP. Hosting a basic server from the ESP32 to control various digital based actuators (led, buzzer, relay) from a simple web page. (2) Displaying various sensor readings on a simple web page hosted on the ESP32
7	**MQTT Based:** (1) Controlling LEDs/Motors from an Android/Web app, Controlling AC Appliances from an android/web app with the help of relay. (2) Displaying humidity and temperature data on a web-based application
8	**UAV / Drone:** (1) Demonstration of UAV elements, Flight Controller (2) Mission Planner flight planning design

CO- PO Mapping:

Semester : 3	**Course Name : Internet of Things**							
	POs							
Course Outcomes	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3	1	-	1	-	-	-	-
CO2	2	2	1	1	-	-	-	-
CO3	2	2	1	2	-	-	-	-
CO4	2	2	3	2	1	1	-	-
CO5	-	3	2	3	-	-	-	-

*Legend: **'3'** for high, **'2'** for medium, **'1'** for low and **'-'** for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094161

Course / Subject Name: Cyber Security and Forensic

w. e. f. Academic Year:	2024-25
Semester:	3
Category of the Course:	Elective Group - 2

Prerequisite:	• Basic knowledge of networking and operating systems • Understanding of computer architecture • Familiarity with programming and scripting (Python preferred)
Rationale:	Cyber security and forensic knowledge is indispensable in the era of increasing cyber threats. This course introduces foundational concepts of cyber security and digital forensics, empowering students to protect, detect, analyze, and respond to cyber incidents using open-source tools. It blends theoretical understanding with practical hands-on lab sessions, enhancing employability in security analysis and forensic investigation roles. **Course Pedagogy:** The course adopts an open-source-centric, hands-on pedagogy. It integrates real-world cyber-attack simulations, digital forensic case studies, and interactive lab sessions. Students will gain proficiency in threat detection, network security, and forensic analysis using tools like Wireshark, Autopsy, Volatility, and Kali Linux. Collaborative and problem-based learning strategies will be employed to simulate real cyber-incident handling.

Course Outcome:

After Completion of the Course, students will be able to:

No	Course Outcomes	RBT Level
01	Explain key concepts of cyber security, vulnerabilities, and threat landscape.	UN
02	Apply open-source tools to perform network traffic analysis and packet capture.	AP
03	Analyze and respond to cyber-attacks using forensic investigation techniques.	AN
04	Use digital forensic tools to recover and examine digital evidence.	AP
05	Evaluate the effectiveness of cyber defense mechanisms and report findings.	EL

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
2	0	2	3	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	Introduction to Cyber Security: Threats, Attacks, Vulnerabilities, CIA triad, Cyber Laws, Kali Linux Overview	9	20
2.	Network Security: Packet analysis using Wireshark, Firewalls, Snort for intrusion detection, DNS spoofing, ARP attacks	9	20
3.	System Security: Linux/Windows security, Malware analysis, Password cracking (John the Ripper, Hashcat), Metasploit	9	20
4.	Digital Forensics: Basics, Chain of custody, Disk imaging (dd, FTK Imager), Hashing, Timeline Analysis	9	20
5.	Forensic Tools and Reporting: Autopsy, Volatility, Registry analysis, Email and browser forensics, Report writing	9	20
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	20	25	15	15	15

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:
(a) Reference Books:

1. William Stallings, "Network Security Essentials", Pearson
2. Nelson, Phillips, Steuart, "Guide to Computer Forensics and Investigations", Cengage
3. Chuck Easttom, "Computer Security Fundamentals", Pearson
4. Marjie T. Britz, "Computer Forensics and Cyber Crime", Pearson

(b) Online Learning Resources and Opensource software and website:

- NPTEL: Cyber Security and Forensics – https://nptel.ac.in/courses/106105197/
- Wireshark: https://www.wireshark.org/
- Kali Linux: https://www.kali.org/
- Autopsy: https://www.autopsy.com/
- Volatility Framework: https://www.volatilityfoundation.org/

- Snort IDS: https://www.snort.org/
- FTK Imager (Free Version): https://accessdata.com/product-download
- Hashcat: https://hashcat.net/
- John the Ripper: https://www.openwall.com/john/

Suggested List of Practical

1. Install and configure Kali Linux virtual environment
2. Perform network packet analysis using Wireshark
3. Scan and detect vulnerabilities using Nmap and OpenVAS
4. Simulate brute-force attack using Hydra and observe logs
5. Analyze a pcap file and report suspicious traffic
6. Crack password hashes using John the Ripper and Hashcat
7. Perform disk imaging and hashing using dd and sha256sum
8. Recover deleted files using Autopsy
9. Analyze memory dump using Volatility
10. Create a digital forensic investigation report

List of Active Learning Assignments:

1. Analyze and document a recent cyber-attack case
2. Perform a comparative analysis of different forensic tools (Autopsy vs Sleuthkit)
3. Simulate a network breach and write a detailed investigation report
4. Create a short video demo of recovering deleted files from a USB drive
5. Present a group-based discussion on the ethical and legal implications of cyber forensics

CO- PO Mapping:

Semester 3	**Course Name : Containerization**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	-	-	-	-	-
CO2	2	3	2	2	-	-	-	-
CO3	2	3	3	3	-	-	-	-
CO4	2	2	3	3	-	-	-	-
CO5	2	2	2	3	-	-	-	-

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *

GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code: MC03094171

Course / Subject Name: Research Work(Phase-1)/Minor Project

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Core Courses

Prerequisite:	• Completion of core courses in the MCA program • Basic understanding of software development life cycle, research methodology, and programming tools • Exposure to domain-specific electives (e.g., AI, Web Development, Mobile Applications, Cloud Computing, etc.)
Rationale:	This phase serves as a foundation for the final semester research project or major software development project. It allows students to identify research problems or software development challenges, perform background research or feasibility studies, define objectives, and propose methodologies. The aim is to foster analytical thinking, domain knowledge, project planning, and ethical considerations in applied research and development. **Pedagogy:** • Problem-based Learning • Self-Directed and Mentor-Guided Research • Regular Review Meetings (Weekly) • Design Thinking Workshops • Collaborative Documentation & Feedback • Presentations and Peer Review • Use of Project Management Tools (e.g., GitHub, Trello, JIRA)

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Identify and define a clear research problem or project challenge aligned with academic and societal needs	AN
02	Conduct literature review or market survey to assess state-of-the-art in chosen area	AP
03	Formulate project objectives, scope, and methodology with clear deliverables	AP
04	Design initial project architecture, model, or prototype as per domain-specific standards	CR
05	Prepare a detailed project proposal including timeline, tools, risks, and resources	AP
06	Demonstrate communication, planning, and presentation skills through interim reviews	AP

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
0	0	6	3	0	0	50	50	100

Guidelines for Student Work:

1. **Project Selection:**
   - Individual or group (max 2-3 students)
   - Should be from chosen domain/elective
   - Must be feasible and implementable within time and resource constraints
2. **Proposal Components:**
   - Title, Abstract, Introduction
   - Problem Statement, Objectives, and Scope
   - Literature Review / Market Survey
   - Proposed Solution/Methodology
   - Tools/Technologies/Hardware
   - Tentative Timeline and Work Plan
   - Expected Outcomes
   - Ethical, Legal, and Environmental Considerations
3. **Milestones:**
   - Week 1–2: Topic Finalization and Approval
   - Week 3–5: Literature Review / Survey
   - Week 6–9: Methodology Development and Planning
   - Week 10–12: Initial Design/Model/Architecture
   - Week 13–14: Final Proposal Document Preparation
   - Week 15: Internal Presentation and Evaluation

Assessment Method:

Component	Weightage	Description
Topic Approval & Synopsis	10%	Relevance, clarity, and originality
Literature/Survey Review	20%	Depth and justification of problem
Interim Review & Progress	20%	Methodology, work done, clarity
Proposal Report	30%	Quality of planning and documentation
Final Presentation & Viva	20%	Delivery, justification, readiness

References/Suggested Learning Resources:
(a) Books:

Reference Books:

1. Ranjit Kumar, Research Methodology: A Step-by-Step Guide for Beginners, Sage Publications
2. C.R. Kothari, Research Methodology: Methods and Techniques, New Age International
3. Ian Sommerville, Software Engineering, Pearson
4. Pressman & Maxim, Software Engineering: A Practitioner's Approach, McGraw-Hill
5. Field-specific journal papers, technical blogs, white papers

(b) Tools & Platforms:

- Reference Managers: Mendeley, Zotero
- Documentation: LaTeX, MS Word
- Collaboration: GitHub, Overleaf, Google Drive
- Project Management: Trello, JIRA, Notion
- Survey: Google Forms, SurveyMonkey

CO- PO Mapping:

Semester 3	**Course Name: Research Work(Phase-1)/Minor Project**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	2	2	1	2
CO2	3	3	2	2	1	1	1	3
CO3	2	3	3	2	2	3	2	2
CO4	3	2	3	3	2	2	1	2
CO5	2	2	3	2	2	3	2	3
CO6	2	2	2	2	3	3	2	3

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *