M.D. UNIVERSITY, ROHTAK

SCHEME OF STUDIES AND EXAMINATION

B. TECH Minor in Computer Science and Engineering

(For ECE and allied branches only)

Scheme effective from 2023-24

(Applicable in UIET, MDU, Rohtak only)

B. TECH Minor in Computer Science and Engineering
(For ECE and allied branches only)
Scheme effective from 2023-24
(Applicable in UIET, MDU, Rohtak only)

Sr. No	Semester	Course No.	Subject	Teaching Schedule			Examination Schedule (Marks)				Duration of Exam (Hours)	No of hours/ week
				L	P	Total Credits	Marks of Internal Assessments	Theory	Practical	Total		
1	3	23CSEEM301	Introduction to operating systems	3	-	3	25	75	-	100	3	3
2	3	23CSEEM302	Data Structures Lab	0	2	1	25	-	25	50	3	2
3	4	23CSEEM401	Introduction to DBMS	3	-	3	25	75	-	100	3	3
4	4	23CSEEM402	DBMS Lab	0	2	1	25	-	25	50	3	2
5	5	24CSEEM501	Introduction to AI and Data Science	3	-	3	25	75	-	100	3	3
6	5	24CSEEM502	Programming in C++ Lab	0	2	1	25	-	25	50	3	2
7	6	24CSEEM601	Network Security	3	_	3	25	75	-	100	3	3
8	6	24CSEEM602	Programming in Python Lab	0	2	1	25	-	25	50	3	2
9	7	25CSEEM701	Introduction to Java Programming	3		3	25	75	-	100	3	3
10	7	25CSEEM702	Java Programming Lab	0	2	1	25	-	25	50	3	2
			TOTAL			20						

Introduction to Operating System

Course code	23CSEEM301				
Category	Minor Degree Course				
Course title	Introduction to Operating System				
Scheme and Credits	L	T	P	Credits	Semester-3
	3	0	0	3	
Internal Assessment	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have parts all units and remaining eight questions of 15 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

UNIT 1:

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services.

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, Multithreading.

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, SRTF, RR Scheduling.

UNIT 2:

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, The Producer\ Consumer Problem, Semaphores, , Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, and Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

UNIT 3:

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation; Paging: Principle of operation – Page allocation –Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures –Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Optimal Page Replacement and Least Recently used (LRU).

UNIT 4:

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), efficiency and performance.
Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

Suggested books:

- Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
- Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Suggested reference books:

- Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
- Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
- Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
- Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Course Outcomes:

- Understand the structure and architectural components of OS to analyze and design the applications to run in parallel. Moreover, students would be able to develop scheduling algorithms to optimize various parameters like CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time for research purposes.

- Understand the design issues associated with Operating systems (e.g. Mutual exclusion, Deadlock detection etc.) to gain insight towards developing algorithms/techniques for efficient deadlock handling.

- For a given specification of memory organization, develop the techniques for optimally allocating memory to processes by increasing memory utilization and improving the access time.

- Design and implement a file management system for a given specification. Identify, use and evaluate the disk management policies with respect to various performance evaluation parameters.

Data Structures Lab

Course code	23CSEEM302				
Category	Minor Degree Course				
Course title	Data Structures Lab				
Scheme and Credits	L	T	P	Credits	Semester-3
	0	0	2	1	
Internal Assessment	25 Marks				
Exam	25 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

Data Structures Lab List of practical exercises, to be implemented using C Language.

1. Write a menu-driven program that implements the following operations (using separate functions) on a linear array:
    - Insert a new element at the end as well as at a given position
    - Delete an element from a given whose value is given or whose position is given
    - To find the location of a given element
    - To display the elements of the linear array
2. Write a menu driven program that maintains a linear linked list whose elements are stored in on ascending order and implements the following operations (using separate functions):
    - Insert a new element
    - Delete an existing element
    - Search an element
    - Display all the elements
3. Write a program to demonstrate the use of stack (implemented using linear array) in converting arithmetic expressions from infix notation to postfix notation.
4. Program to demonstrate the use of stack (implemented using linear linked lists) in evaluating arithmetic expression in postfix notation.
5. Program to demonstrate the implementation of various operations on a linear queue represented using a linear array.
6. Program to demonstrate the implementation of various operations on a circular queue represented using a linear array.
7. Program to demonstrate the implementation of various operations on a queue represented using a linear linked list (linked queue).
8. Program to illustrate the implementation of different operations on a binary search tree.
9. Program to illustrate the traversal of graph using breadth-first search
10. Program to illustrate the traversal of graph using depth-first search.
11. Program to sort an array of integers in ascending order using bubble sort.
12. Program to sort an array of integers in ascending order using selection sort.
13. Program to sort an array of integers in ascending order using insertion sort.
14. Program to sort an array of integers in ascending order using radix sort.
15. Program to sort an array of integers in ascending order using merge sort.
16. Program to sort an array of integers in ascending order using quick sort.

17. Program to sort an array of integers in ascending order using heap sort.
18. Program to sort an array of integers in ascending order using shell sort.
19. Program to demonstrate the use of linear search to search a given element in an array.
20. Program to demonstrate the use of binary search to search a given element in a sorted array in ascending order.

INTRODUCTION TO DBMS

Course code	23CSEEM401				
Category	Minor Degree Course				
Course title	INTRODUCTION TO DBMS				
Scheme and Credits	L	T	P	Credits	Semester 4
	3	0	0	3	
Internal Assessment	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have parts all units and remaining eight questions of 15 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Objectives of the course

a. To understand the different issues involved in the design and implementation of a database system.

b. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models

c. To understand and use data manipulation language to query, update, and manage a database

d. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.

e. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

UNIT-I

Introduction: Basic Definitions and Concept, Traditional file System Versus Database Systems, DBMS Users, Database or DBMS Languages, Schemes, Sub-schema and Instances, Three Level Architecture of Database Systems (DBMS), Client-Server Architecture, **Data Models**: Introduction, Hierarchal Model, Network Model, Relational Model, Comparison of DBMS and RDBMS, Comparison of data models

E-R Model : Introduction, Basic Concepts, Types of Attributes, Relationship Sets, Mapping Constraints, Keys, Entity—Relationship Diagram, Types of Entity Sets.

UNIT-II

Relational Algebra and calculus: Introduction, Relational algebra, Relational calculus, Comparison of Domain relational calculus and Tuple relational calculus, Comparison of relational calculus and relational algebra. **Query language**: Introduction, Structured Query language (SQL)

Functional dependencies and normalizations: Introduction, Informal design guidelines for relational schemas, Functional dependencies, Anomalies in relational database, Normalization

UNIT-III

Transactions and Concurrency Control: Introduction, Transaction, Some Definitions, Why Concurrency Control is Needed, Concurrency Control Techniques, Deadlocks,

Database Security and Authorization: Introduction, Security Violations, Authorization, Views.

UNIT-IV

Query Processing and Optimization: Introduction, Basics of Query Processing, Query Optimization

Data Warehouse and mining: Introduction, Data mining, Comparison of data mining and data warehousing.

Suggested books:

"Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

Suggested reference books

"Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.

"Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education

"Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

Course Outcomes

1. For a given query write relational algebra expressions for that query and optimize the developed expressions
2. For a given specification of the requirement, design the databases using E R method and normalization.
3. For a given specification, construct the SQL queries for Open source and Commercial DBMS -MYSQL, ORACLE, and DB2.
4. For a given query optimize its execution using Query optimization algorithms
5. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.

DBMS Lab

Course code	23CSEEM402				
Category	Minor Degree Course				
Course title	DBMS Lab				
Scheme and Credits	L	T	P	Credits	Semester 6
	0	0	4	2	
Internal Assessment	25 Marks				
Exam	25 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

Course Objectives:

- Keep abreast of current developments to continue their own professional development
- To engage themselves in lifelong learning of Database management systems theories and technologies which enables them to pursue higher studies.
- To interact professionally with colleagues or clients located abroad and the ability to overcome challenges that arise from geographic distance, cultural differences, and multiple languages in the context of computing.
- Develop team spirit, effective work habits, and professional attitude in written and oral forms, towards the development of database applications.

1. To study the basics of Structured Query Language(SQL)
2. To create table and insert values in it
3. To study the use of alter, describe and drop statements
4. To study the use of delete and update statements
5. To study the various query processing statements
6. To study the join operations
7. To study the set operations
8. SQL Commands

Introduction to AI and Data Science

Course code	24CSEEM501				
Category	Minor Degree Course				
Course title	Introduction to AI and Data Science				
Scheme and Credits	L	T	P	Credits	Semester 6
	3	0	0	3	
Internal Assessment	25 Marks				
Exam (Semester Exam)	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have parts all units and remaining eight questions of 15 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Objectives of the course

1. Demonstrate an understanding of statistics and machine learning concepts that are vital for data science;
2. Produce Python code to statistically analyse a dataset;
3. Learn the basic AI approaches
4. Develop problem solving agents

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have parts from all units and the remaining eight questions of 15 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

UNIT I

Introduction to Data Science, Different Sectors using Data science, Purpose and Components of Python in Data Science. Applications of Data Science, Data Science and Ethical Issues- Discussions on privacy, security, ethics- A look back at Data Science- Next-generation data scientists.

UNIT II

Data Analytics Process, Knowledge Check, Exploratory Data Analysis (EDA), EDA-Quantitative technique, EDA-Graphical Technique, Data Analytics Conclusion and Predictions.

UNIT III

Introduction to AI – Agents and Environments – concept of rationality – nature of environments – structure of agents. Problem solving agents – search algorithms – uninformed search strategies.

UNIT IV

Heuristic search strategies – heuristic functions. Local search and optimization problems – local search in continuous space – search with non-deterministic actions – search in partially observable environments – online search agents and unknown environments

Outcomes of the course

1. To explain how data is collected, managed and stored for data science;
2. To understand the key concepts in data science, including their real-world applications and the toolkit used by data scientists;
3. Explain intelligent agent frameworks CO2:
4. Apply problem solving techniques

Text Books:

1. Data Sciences & Analytics, V.K. Jain, Khanna Publishing House.

2. Business Analytics: The Science of Data - Driven Decision Making, U Dinesh Kumar, John Wiley & Sons.
3. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, John Wiley & Sons.
4. Joel Grus, Data Science from Scratch, Shroff Publisher/O'Reilly Publisher Media

Reference Books:

1. Annalyn Ng, Kenneth Soo, Numsense! Data Science for the Layman, Shroff Publisher Publisher
2. Cathy O'Neil and Rachel Schutt. Doing Data Science, Straight Talk from The Frontline. O'Reilly Publisher.
3. Jure Leskovek, Anand Rajaraman and Jeffrey Ullman. Mining of Massive Datasets. v2.1, Cambridge University Press.
4. Jake VanderPlas, Python Data Science Handbook, Shroff Publisher/O'Reilly Publisher Media.
5. Philipp Janert, Data Analysis with Open Source Tools, Shroff Publisher/O'Reilly Publisher Media.

Programming in C++ Lab

Course code	24CSEEM502				
Category	Minor Degree Course				
Course title	Programming in C++ Lab				
Scheme and Credits	L	T	P	Credits	Semester 6
	0	0	2	1	
Internal Assessment	25 Marks				
Exam	25 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

Contents:

1. [Classes and Objects] Write a program that uses a class where the member functions are defined inside a class.
2. [Classes and Objects] Write a program that uses a class where the member functions are defined outside a class.
3. [Classes and Objects] Write a program to demonstrate the use of static data members.
4. [Classes and Objects] Write a program to demonstrate the use of const data members.
5. [Constructors and Destructors] Write a program to demonstrate the use of zero argument and parameterized constructors.
6. [Constructors and Destructors] Write a program to demonstrate the use of dynamic constructor.
7. [Constructors and Destructors] Write a program to demonstrate the use of explicit constructor.
8. [Initializer Lists] Write a program to demonstrate the use of initializer list.
9. [Operator Overloading] Write a program to demonstrate the overloading of increment and decrement operators.
10. [Operator Overloading] Write a program to demonstrate the overloading of binary arithmetic operators.
11. [Operator Overloading] Write a program to demonstrate the overloading of memory management operators.
12. [Inheritance] Write a program to demonstrate multilevel inheritance.
13. [Inheritance] Write a program to demonstrate multiple inheritance.
14. [Inheritance] Write a program to demonstrate the virtual derivation of a class.
15. [Polymorphism] Write a program to demonstrate the runtime polymorphism.
16. [Exception Handling] Write a program to demonstrate exception handling.
17. [Templates and Generic Programming] Write a program to demonstrate the use of function template.
18. [Templates and Generic Programming] Write a program to demonstrate the use of class template.

NETWORK SECURITY

Course code	24CSEEM601				
Category	Professional Elective Course				
Course title	Network Security				
Scheme and Credits	L	T	P	Credits	Semester 6
	3	0	0	3	
Internal Assessment	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have parts all units and remaining eight questions of 15 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Course Objectives:

1. To understand cryptography theories; algorithms & systems.
2. To understand the symmetric and asymmetric key algorithms.
3. To understand necessary approaches & techniques to build protection mechanisms in order to secure Computer Networks.
4. Acquire fundamental knowledge on the concepts of different security layers.

UNIT- I

Introduction: Plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography.

UNIT- II

Symmetric Key Algorithms:- Introduction, algorithms types and modes, DES, AES.

Asymmetric Key Algorithms: Introduction, history of asymmetric key cryptography, RSA symmetric and asymmetric key cryptography together, Digital signature.

UNIT- III

Internet Security Protocols: Basic concepts, Secure Socket Layer (SSL), Transport Layer Security (TLS), Secure Hyper Text Transfer protocol (SHTTP), Time Stamping Protocol (TSP), Secure Electronic Transaction (SET), S SL versus SET, Electronic Money, Email Security.

UNIT- IV

User Authentication And Kerberos:- Introduction, Authentication basics, Passwords, authentication tokens, certificate based authentication, biometric based authentication, Kerberos, key distribution center(KDC), Security handshake pitfalls, single Sign on(SSO) approach.

TEXT/ REFERENCE BOOKS:

1. Cryptography and Network Security, 2nd Edition by Atul Kahate, TMH
2. Network Management Principles & Practices by Subramanian, Mani (AWL)
3. SNMP, Stalling, Willian (AWL)
4. SNMP: A Guide to Network Management (MGH)
5. Telecom Network Management by H.H. Wang (MGH)
6. Network Management by U. Dlack (MGH)

Course Outcomes:

After completing the course the student will be able to

1. Compare various cryptographic techniques.
2. Work with symmetric & asymmetric key algorithms.
3. Design secure applications.
4. Inject secure coding in the developed applications.

Programming in Python Lab

Course code	24CSEEM602				
Category	Minor Degree Course				
Course title	Programming in Python Lab				
Scheme and Credits	L	T	P	Credits	Semester- 6
	0	0	2	1	
Internal Assessment	25 Marks				
Exam	25 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

Objectives

- To write, test, and debug simple Python programs.
- To implement Python programs with conditionals and loops.
- Use functions for structuring Python programs.
- Represent compound data using Python lists, tuples, and dictionaries.
- Read and write data from/to files in Python.

List of Programs

1. Compute the GCD of two numbers.
2. Find the square root of a number (Newton‘s method)
3. Exponentiation (power of a number)
4. Find the maximum of a list of numbers
5. Linear search and Binary search
6. Selection sort, Insertion sort
7. Merge sort
8. First n prime numbers
9. Multiply matrices
10. Programs that take command line arguments (word count)
11. Find the most frequent words in a text read from a file
12. Simulate elliptical orbits in Pygame
13. Simulate bouncing ball using Pygame

Outcome:

- Write, test, and debug simple Python programs.
- Implement Python programs with conditionals and loops
- Develop Python programs step-wise by defining functions and calling them.
- Use Python lists, tuples, dictionaries for representing compound data.

INTRODUCTION TO JAVA PROGRAMMING

Course code	25CSEEM701				
Category	Minor Degree Syllabus				
Course title	Introduction to Java Programming				
Scheme and Credits	L	T	P	Credits	Semester 7
	3	0	0	3	
Internal Assessment	25 Marks				
Exam	75 Marks				
Total	100 Marks				
Duration of Exam	03 Hours				

Course Objectives:

- Programming in the Java programming language.
- Knowledge of object-oriented paradigm in the Java programming language.
- The use of Java in a variety of technologies and on different platforms.

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have parts all units and remaining eight questions of 15 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit 1:

Introduction to Java: Evolution of Java, Object Oriented Programming Structure, Overview and characteristics of Java, Java program Compilation and Execution Process, Organization of the Java Virtual Machine, Client side Programming, Platform Independency & Portability, Security, Relation b/w JVM, JRE and JDK, Introduction to JAR format, Naming Conventions, Data types & Type casting, operators, Security Promises of the JVM, Security Architecture and Security Policy, security aspects, sandbox model

Unit 2:

OOPS Implementation: Classes, Objects, attributes, methods, data encapsulation, reference variables, Constructors, Anonymous block, Method Overloading, Static Data members, Block & methods; Memory Structure: Stack, Heap, Class & Method area

Class loading & Execution flow: Static vs Dynamic Class loading, implicit vs explicit class loading, class loading operations;

Argument Passing Mechanism: Passing primitive arguments, passing objects, Wrapper Classes;

This keyword: Referencing instance members, Intra class constructor chaining, Method chaining;

Inheritance & code reusability: Extending classes for code reusability, Usage of super keyword, Method Overriding, Object class

Unit 3:

Inheritance & Runtime Polymorphism: Static & Dynamic binding, Inheritance and Is-A relation, Runtime Polymorphism and Generalization, Abstract classes & methods, Final Keyword;
Interfaces and Role based Inheritance: Feature & Role based Inheritance, Static & Dynamic classing Environment, classes & interfaces, interface applications in real scenarios; Has-A relation: Aggregation & Composition, Nested classes, Inner classes, Anonymous Inner classes, String Buffer Class, tokenizer, applets, Life cycle of applet and Security concerns

Unit 4:

Package & Scopes: Need of Packages, associating classes to Packages, Class path environment variable, Import Keyword and Feature of static import, Public, protected, private & default scope, Private Inheritance; **Exception Handling:** exception and error, Exception Handling & Robustness, Common Exceptions and Errors, Try and catch block, Exception handlers, throw keyword, Checked and Unchecked Exceptions, Role of finally, User defined Exceptions;

Text Books:

1. Patrick Naughton and HerbertzSchidt, “Java-2 the complete Reference”, TMH
2. Sierra & bates, “Head First Java”, O’Reilly.

Reference Books:

1. E. Balaguruswamy, “Programming with Java”, TMH
2. Horstmann, “Computing Concepts with Java 2 Essentials”, John Wiley.
3. Decker &Hirshfield, “Programming.Java”, Vikas Publication.

Course Outcomes:

- Knowledge of the structure and model of the Java programming language, (knowledge)

Java Programming Lab

Course code	25CSEEM702				
Category	Minor Degree Course				
Course title	Java Programming Lab				
Scheme and Credits	L	T	P	Credits	Semester 7
	0	0	2	1	
Internal Assessment	25 Marks				
Exam	25 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

List of Experiments:

1. Create a java program to implement stack and queue concept.
2. Write a java package to show dynamic polymorphism and interfaces.
3. Write a java program to show multithreaded producer and consumer application.
4. Create a customized exception and also make use of all the 5 exception keywords.
5. Convert the content of a given file into the uppercase content of the same file.
6. Develop an analog clock using applet.
7. Develop a scientific calculator using swings.
8. Create an editor like MS-word using swings.
9. Create a servlet that uses Cookies to store the number of times a user has visited your servlet.
10. Create a simple java bean having bound and constrained properties.