DEBRA THANA SAHID KSHUDIRAM SMRITI MAHAVIDYALAYA

Gangaram Chak, Chak Shyampur, Debra, West Bengal

PROPOSED SYLLABUS (DRAFT) OF

BACHELOR OF SCIENCE WITH COMPUTER SCIENCE (MULTIDISCIPLINARY STUDIES)

3-YEAR UNDERGRADUATE PROGRAMME

(w.e.f. Academic Year 2025-2026)

Based on

Curriculum & Credit Framework for Undergraduate Programmes (CCFUP), 2023 & NEP, 2020

Level	YR.	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks		
								CA	ESE	TOTAL
B.Sc. in Physical Sc./ Math. & Comp. Sc. with Computer Science	3rd	V	**SEMESTER-V**							
			Major (Disc.-A4)	UG/V/COMP/3/MJ-A4	T: Digital Logic Design; P: Practical *(To be studied by the students taken Computer Science as Discipline-A)*	4	3-0-1	15	60	75
			Major (Disc.-A5)	UG/III/COMP/3/MJ-A5	T: Design and Analysis of Algorithms ; P: Practical *(To be studied by the students taken Computer Sc. as Discipline- A)*	4	3-0-1	15	60	75
			Major (Disc.-A6)	UG/III/COMP/3/MJ-A6	T: Database Management Systems; P: Practical *(To be studied by the students taken Computer Sc. as Discipline- A)*	4	3-0-1	15	60	75
			Major (Elective) -2	UG/III/COMP/3/MJE-2	T: Java Programming/ .NET; P: Practical *(To be studied by the students taken Computer Sc. as Discipline- A)*	4	3-0-1	15	60	75
			Minor (Disc.-C5)	UG/III/COMP/3/MI-C5	T: Operating System; P: Practical (To be studied by students taken Computer Sc. as Discipline- C)	4	3-0-1	15	60	75
					Semester-V Total	**20**				**375**
		VI	**SEMESTER-VI**							
			Major (Disc.-B4)	UG/IV/COMP/3/MJ-B3	T: Digital Logic Design; P: Practical *(Same as like A2 for students taken Computer Science as Discipline-B)*	4	3-0-1	15	60	75
			Major (Disc.-B5)	UG/IV/COMP/3/MJ-B5	T: Design and Analysis of Algorithms ; P: Practical *(To be studied by the students taken Computer Sc. as Discipline- B)*	4	3-0-1	15	60	75
			Major (Disc.-B6)	UG/IV/COMP/3/MJ-B6	T: Database Management Systems; P: Practical *(To be studied by students taken Computer Sc. as Discipline- A)*	4	3-0-1	15	60	75
			Major (Elective) -3	UG/III/COMP/3/MJE-3	T: Machine Learning; P: Practical *(To be studied by the students taken Computer Sc. as Discipline- A)*	4	3-0-1	10	40	50
			Minor (Disc.-C6)	UG/IV/COMP/3/MI-C6	T: Database Management System; P: Practical *(To be studied b the students taken Computer Sc. as Discipline- C)*	4	3-0-1	15	60	75
					Semester-VI Total	**20**				**375**
					TOTAL of YEAR-3	**40**	**-**	**-**	**-**	**700**
			Eligible to be awarded Bachelor of Science in Multidisciplinary Studies with Computer Science on Exit			**126**	**Marks (Year: I+II+III)**			**2325**

MJ= Major Programme (Multidisciplinary), MN = Minor, A/B = Choice of Major Discipline; C= Choice of Minor Discipline; SEC = Skill Enhancement Course, AEC = Ability Enhancement Course, MDC = Multidisciplinary Course, VAC = Value Added Course; CA= Continuous Assessment, ESE= End Semester Examination, T = Theory, P= Practical, L-T-P = Lecture-Tutorial-Practical, MIL = Modern Indian Language, ENVS= Environmental Studie

SEMESTER-V

UG/V/COMP/3/MJ-A4: Digital Logic Design Credits 04

Objective of the course:

• Understanding of positional number systems, including Binary, Octal, Hexadecimal, and Decimal, and learn how to perform conversions between them.
• Grasp the techniques for representing signed numbers, including signed magnitude, one's complement, and 2's complement, and understand their significance in digital circuits.
• Develop proficiency in performing binary arithmetic operations.
• Utilize basic logic gates (OR, AND, NOT) and universal logic gates (NAND & NOR) to implement logical operations and construct complex digital circuits.
• Perform logical operations including logical sum (OR), logical product (AND), complementation (NOT), Anti-coincidence (EX-OR), and Coincidence (EX-NOR), enabling the manipulation of digital signals.
• Master the techniques for simplifying Boolean expressions using algebraic methods and Karnaugh map methods, facilitating the optimization of digital circuits.
• Design a variety of digital circuits, including Half Adder, Full Adder, Half-Subtractor, Full Subtractor, Multiplexers, Encoders, De-multiplexers, Decoders.
• Understand the principles of flip-flops including latch, RS, D, JK, T Flip Flops.

Course Contents:

MJ-A4T: Digital Logic Design Theory Credits 03

Module I: Number Systems (10 Hours)

Introduction to positional number systems: Binary, Octal, Hexadecimal, and Decimal.
Binary arithmetic operations: addition, subtraction, multiplication, and division. Number system conversions: binary to decimal, decimal to binary, binary to octal, octal to binary, binary to hexadecimal, and hexadecimal to binary, BCD (Binary Coded Decimal).
Signed number representation: signed magnitude, one's complement, and 2's complement.

Module II: Boolean Algebra (10 Hours)

Fundamentals of Boolean algebra: Boolean expressions, Boolean operators (AND, OR, NOT), and Boolean laws. Basic logic gates: AND gate, OR gate, NOT gate. Universal logic gates: NAND gate, NOR gate. Exclusive gates: Ex-OR, Ex-NOR. Simplification of Boolean expressions using algebraic manipulation, truth tables, and Karnaugh maps.

Module III : Combinational Circuits (13 Hours)

Design of basic combinatorial logic circuits: Half Adder, Full Adder, Half-Subtractor, Full-Subtractor. Multiplexers: design, applications, and implementation. De-multiplexers: design, applications, and implementation. Encoders: design, applications, and implementation. Decoders: design, applications, and implementation. Seven Segment Display: design, applications, and implementation. BCD adder/subtractor: design, applications, and implementation.

Module IV: Sequential Circuits **(12 Hours)**

Introduction to sequential logic circuits: latch, RS flip-flop, D flip-flop, JK flip-flop, T flip-flop. Edge Triggered Flip-Flops. Master-Slave JK Flip Flop: design, implementation, and analysis. Registers: Serial Input Serial Output (SISO), Serial Input Parallel Output (SIPO), Parallel Input Serial Output (PISO), Parallel Input Parallel Output (PIPO), Universal Shift Registers. Asynchronous Counters: design, applications, and implementation. Synchronous Counters: design, applications, and implementation.

MJ-A4P: Digital Logic Design Lab **Credits: 01**

1. Experiment of fundamental logic gates: AND, OR, NOT.
2. Implementation of different functions using Basic and Universal Logic gates.
3. Implementation of Basic gates using NAND and NOR gates.
4. Design and implement a Half Adder circuit.
5. Design and implement a Full Adder circuit.
6. Design and implement a Half-Subtractor circuit.
7. Design and implement a Full-Subtractor circuit.
8. Program to design and implement a Multiplexer.
9. Program to design and implement a De-multiplexer.
10. Program to design and implement an Encoder.
11. Program to design and implement a Decoder.
12. Design and implement an RS flip-flop.
13. Design and implement a D flip-flop.
14. Design and implement a JK flip-flop.
15. Design and implement a T flip-flop.
16. Program to design and implement a master-slave JK flip-flop.

Reference Books For Digital Logic Design

1. Malvino and Leach, “Digital Principles and Applications”.
2. R.P. Jain,“Modern Digital Electronics” .
3. S.Salivahanan, S.Arivazhagan ,“Digital Circuits & Design” — Vikas Publishing House Pvt Ltd.
4. M.Mano, “Digital logic & Computer Design“,Prentice Hall of India.
5. A. Annand Kumar ,“Fundamental of Digital Circuits”.
6. David Harris and Sarah Harris ,"Digital Design and Computer Architecture".
7. Guy Even and Moti Medina ,"Digital Logic Design: A Rigorous Approach" by
8. Ronald J. Tocci, Neal S. Widmer, and Greg Moss ,"Digital Systems: Principles and Applications".

UG/V/COMP/3/MJ-A5: Design and Analysis of Algorithms Credits 04

Objective of the course:

- Equip students with the skills to create efficient algorithms and analyze their performance.
- Cover fundamental algorithmic techniques such as divide and conquer, dynamic programming, and greedy algorithms.
- Teach students to analyze time and space complexity of algorithms using Big O notation.

- Emphasize problem-solving strategies for designing optimal solutions to computational problems.
- Study data structures and their impact on algorithm efficiency.
- Provide practical experience through programming assignments and projects.
- Explore advanced topics like NP-completeness and approximation algorithms for addressing complex problems.
- Prepare students to evaluate and develop efficient algorithms, laying the foundation for careers in software development, data analysis, and research in computer science.

Course Contents:

MJ-A5T: Design and Analysis of Algorithm Theory **Credits 03**

Module I: Basics of an Algorithm **(15 Hrs.)**

Definition, characteristics and steps in designing of algorithms, Asymptotic notations (O, Ω, Θ etc.), Recursion(Hanoi problem, Tail Recursion.) and Recurrence relation, Substitution Method, Iteration Method, Master Method. Concept of efficiency of analysis of an algorithm (Linear Search, Insertion Sort, Selection sort.) Comparative efficiencies of algorithms: Linear, Quadratic, Polynomial and Exponential.

Module II: Divide and Conquer Approach **(08 Hrs.)**
General Issues in Divide and Conquer, Binary Search, Merge Sort, Quick Sort.

Module III: Greedy Technique **(08 Hrs.)**
Elements of Greedy strategy, Knapsack Problem, single-source shortest path (Dijkstra's algorithm.

Module IV: Dynamic Programming **(08 Hrs.)**
Basic method, use, Examples: All pair shortest paths, single source shortest path (Bellman Ford), Traveling Salesman problem.

Module V: Graph Algorithm **(06 Hrs.)**
Representation of Graphs, Adjacency Matrix, Adjacency List, Depth First Search and Examples, Breadth First Search and Examples, Spanning Tree(Prims, Kruskal's).

References:
1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2. Sarabasse& A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999

MJ-A5P: Design and Analysis of Algorithm Lab **Credits: 01**

1. Implement linear search, Selection Sort, Insertion Sort, Merge Sort, Quick Sort.
2. Implement Breadth-First Search in a graph
3. Implement Depth-First Search in a graph

4. Write a program to determine the minimum spanning tree of a graph
5. Write a program to find minimum cost spanning tree using Kruskal's Algorithm.
6. Write a Program to perform Binary Search for a given set of integer values recursively and non-recursively.
7. Write a program to find solution for knapsack problem using greedy method.

UG/V/COMP/3/MJ-A6: Database Management Systems Credits 04

Objective of the Course:

- Provide a comprehensive understanding of database design, implementation, and Management.
- Cover fundamental concepts of database architecture, data models, and relational database management systems (RDBMS).
- Teach students to design and normalize databases to ensure data integrity and eliminate redundancy.
- Emphasize the use of SQL for querying and manipulating databases, including creating, updating, and deleting records.
- Explore advanced topics such as transactions, concurrency control, and database security for reliable data handling.
- Offer hands-on experience through projects involving the design and implementation of database systems.
- Equip students with skills to design, manage, and optimize databases.
- Prepare students for careers in database administration, data analysis, and software development.

Course Contents:

MJ-A6T: Database Management Systems Theory Credits 03

Module- I:Introduction (6 Hrs.)

Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Data Abstraction, Three Schema architecture of DBMS.

Module- II: Entity Relationship (ER) Model (6 Hrs.)

Entity Set, Simple and composite Attribute, Single valued and multivalued attribute, Relationship sets, Mapping cardinality, keys, , Entity Relationship Diagram : Need for E-R Model, Various steps of database design, Mapping Constraints, E-R diagram, Subclass, Generalization, Specialization, Aggregation, Strong Entity-Weak Entity.

Module- III: Relational Algebra (10 Hrs.)

Select operation, Project Operation, Set operations (union, intersection, difference), Join operations, Division operation, outer join and outer union, Examples queries in Relational Algebra.

Module- IV: SQL (8 Hrs.)

Concept of DDL, DML. Basic Structure Relational databases and tables, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Sub queries.

Module- V: Relational Model and Relational Database Design (10 Hrs.)

Concept of Relational Model, Design Issues, Keys, Closure set, Functional Dependency, Different anomalies in designing a Database., Normalization using functional dependencies, Decomposition, 3NF, Boyce-Codd Normal Form.

Module- VI: Transaction Management (10 Hrs.)

ACID properties, Transaction definition, properties, transaction state diagram, commit and rollback, Serializability (Conflict and View), Concurrency control, lock based protocols, Two phase locking, Timestamp ordering protocol, Recovery management, Deadlock handling and prevention.

References:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, DatabaseSystem Concepts 6th Edition, McGraw
Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

MJ-A6P: Database Management Systems (Practical) Credits 01

1. An inventory database has the following tables:

ITEM (Item_Code, Item_Name, Price)

PURCHASE (Item_Code, Quantity, Purchase_date)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

a. Create the tables with the above attributes and enter 5-7 tuples into each table.
b. Display the List of all items with their price which have minimum 10 quantity.
c. Display the List of items which are not purchased by anyone.
d. Display all the item with their Purchase date.
e. Display all items according to their price.

2. Create the following tables with the fields given below:

TEACHER (T_ID, Department, Year of Exp, Name)

SUBJECT (Sub_Paper_ID, T_ID, Title_of Paper, Programme, Semester)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

a. Create the tables with the above attributes and enter 5-7 tuples into each table.
b. Display Name and Year-Of-Exp of all the teachers of "Computer Science" department.
c. List the Subject Paper which are handled by T_ID = 101.
d. List the name of Programme and Semester of paper titled "C Programming".
e. List the name of teacher who are allotted for 4th semester.

3. Create the following tables with the fields given below:

PRODUCT (Product_ID, Supplier_ID, Proudct_Name, Category, Price, Quantity)

SUPPLIER (Supplier_ID, S_Name, Area, City)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

a. Create the tables with the above attributes and enter 5-7 tuples into each table.
b. List price and product name of all the products whose Quantity is more than 100.
c. Display product name and product ID of the products which are supplied by Supplier_iD = 10111.
d. Display the number of products which price is more than Rs. 1,500 and Quantity is more than 5.
e. Display name and area of all the suppliers of city "Kolkata".

4. Create the following tables with the fields given below:

DOCTOR(Did, DName, Dept_name, Salary, Joining_date)

PATIENT(Pid, Did, Pname, P_address, Phone_no, Admitted_dept)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

a. Create the tables with the above attributes and enter 5-7 tuples into each table.
b. Find the list of patients admitted in “Orthopedic” department.
c. Display the name, Department and Salary of doctor who get maximum salary.
d. Display the total number of patients admitted under “Dr. M Chatterjee”.
e. List the number of patients in each Department.

5. Create the following tables with the fields given below:

STUDENT (Student ID, Name, Programme, Teacher ID)

TEACHER (Teacher ID, Department, Name, City, Specialization)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

a. Create the tables with the above attributes and enter 5-7 tuples into each table.
b. Display total number of students registered in "MCA" programme.
c. Display name of all the students who are associated with TeacherID = 123.
d. Display name, specialization and city of all the teachers who are associated with "Compute Science" department.
e. Display unique Department from the Teacher Table.

6. Create the following tables with the fields given below:

Depertment (Dept_no, D_name, loc)

Employee (Empno, E_name, job, mgr, hiredate, sal, comm, Dept_no)

Select appropriate e primary keys and foreign keys. Select appropriate data types for all the fields.

a. Create the tables with the above attributes and enter 5-7 tuples into each table.
b. Update the employee salary by 15%, whose experience is greater than 10 years.
c. Display the manager who is having maximum number of employees working under him.
d. Display Name of all the employees where the third letter of their name is _A‘.
e. Display Unique Listing of all Jobs that are in Department no 30.

UG/V/COMP/3/MJE-02: Java Programming/ .NET Credits 04

Objective of the course

- Understand the Java environment, JVM architecture, features, and differences from C++, and develop basic programs using variables, data types, operators, control structures, and methods.
- Apply concepts of arrays, strings, and input/output streams to develop programs for data handling, manipulation, and file operations.
- Demonstrate object-oriented programming concepts by designing and implementing classes, constructors, methods, and memory management using garbage collection.
- Implement advanced OOP features such as inheritance, interfaces, packages, polymorphism, and use wrapper classes, enumerations, and metadata effectively.
- Develop robust and efficient Java applications using exception handling, multithreading,

Course Contents

MJE-02T: Programming in Java (Theory) Credits 03

Module -1: **Introduction to Java (8 Hrs.)**

Java Environment: Features, Editions, Architecture, JVM, JDK, JRE, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Command Line Arguments,
Constants, Variables, Data Types: Constants, Variables, Data Types, Declaration of Variables, Giving values to Variables, Symbolic Constants, Type casting.
Operators & Expressions: Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment & Decrement operators, conditional operators, Bitwise Operators, Arithmetic Expressions, Evaluation of Expressions, Type Conversions in Expressions, Operator Precedence & Associativity.
Decision Making, Branching, Looping: Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Built-in Java Class Methods).

Module -2: **Object-Oriented Programming Overview (10 Hrs.)**

Object-Oriented Programming concepts: Principles and features of Object-Oriented Programming, Class, Objects, Encapsulation, Polymorphism, Inheritance, Abstraction.
Class and objects: Define and Using Class and objects, Controlling Access to Class Members (Variables & Methods), Application of two classes.
Methods: Method declaration, Overloading, Use of Constructors, Final keyword, static keyword, this keyword, Abstract keyword and Garbage Collection.

Module -3: **Inheritance, Interfaces, Packages, Wrapper Class (10 Hrs.)**

Inheritance: Inheritance (Single Level, Multilevel and Hierarchical), Method Overriding, Dynamic Method Dispatch, Abstract Classes.
Interfaces: Defining Interface, Extending and Implementing Interface, Multiple Inheritance, Abstraction.
Packages: Basics of packages, System packages (util, lang, io, net), Creating and accessing packages, creating user defined packages, Adding class to a package.
Wrapper Classes: Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata

Module -4: **Arrays, Array List and Strings (8 Hrs.)**

Arrays: Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically.

Array List: Creating and Accessing List, Array List Class, List Interface.

String: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes, String Builder Classes.

Module -5: **Exception Handling, Multithreading** **(8 Hrs.)**

Exception Handling: Using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, Multiple catch statements, Creating user defined exceptions.

Multithreading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads.

Reference Books

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.

2. James Gosling, Bill Joy, Guy L Steele Jr, Gilad Bracha, Alex Buckley "The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.

3. Joshua Bloch, "Effective Java" 2nd Edition, Publisher: Addison-Wesley, 2008.

4. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 1 ,9th Edition, Printice Hall.2012

5. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, Printice Hall.2013

6. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.

7. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.

8. Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011.

9. "Head First Java", Orielly Media Inc. 2nd Edition, 2005.

10. David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.

11. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition, 2004.

MJE-02P: Programming in Java Lab (Practical) **Credits 01**

1. Write a program to find the sum of two integers using command line arguments.
2. Write a program to swap two numbers as input from the keyboard.
3. Write a program to find the factorial of a given number.
4. Write a program to print first n Fibonacci numbers.
5. Write a program to check a number is prime or not.
6. Write a program to find sum of prime factors of given number

7. Write a program to find the reverse of a given number and check it is palindrome number or not.
8. Write a program to print Armstrong number between a given range entered from keyboard.

9. Program to demonstrate method overloading by creating multiple add() methods with different parameter types and arguments to perform addition and show compile-time polymorphism.

10. Program to demonstrate constructor overloading by defining multiple constructors in a class to initialize objects with different values.

11. Program to demonstrate static method by creating a utility method (like finding square or cube) that can be called without creating an object.

12. Program to demonstrate single inheritance by creating a base class (e.g., Animal) and a derived class (Dog) to reuse properties and methods.

13. Program to demonstrate multilevel inheritance using three classes (Grandparent, Parent, Child) to show inheritance across multiple levels.

14. Program to demonstrate hierarchical inheritance where multiple child classes inherit from a single parent class and use its properties.

15. Program to demonstrate method overriding by redefining a method in the child class to provide its specific implementation (runtime polymorphism).

16. Program to demonstrate use of **super** keyword to access parent class constructor and methods from the child class.

17. Program to demonstrate both method overloading and method overriding together to show compile-time and runtime polymorphism in a single application.

18. Program to demonstrate use of package by creating a user-defined package, defining a class inside it, and accessing that class from another program using import statement to show modular programming and code reusability.

19. Program to find the sum and average of elements in an array by taking input values and performing arithmetic operations.
20. Program to check whether a given string is a palindrome by comparing it with its reverse.
21. Program to demonstrate basic string functions like length(), toUpperCase(), toLowerCase(), and trim(),append (), insert (), concat ()and equals () on a given string.
22. Program to compare two strings using equals() and compareTo() methods to check equality and lexical order.
23. Program to search a character or substring in a string using indexOf() and lastIndexOf() methods.
24. Program to demonstrate exception handling using try-catch-finally block by handling arithmetic exception (like division by zero) and ensuring program execution continues normally.
25. Program to demonstrate user-defined exception by creating a custom exception class and throwing it using *throw* and handling it using *catch***.**

26. Program to demonstrate multithreading by creating threads using *Thread class* and *Runnable interface* to execute multiple tasks simultaneously.

.NET Framework

Objective of the course

This course aims to provide students with a comprehensive understanding of the .NET framework and its architecture, including the Common Language Runtime (CLR), Common Type System (CTS), and .NET class libraries. It focuses on enabling learners to develop applications using multiple .NET-supported languages such as C# and VB.NET, while gaining practical knowledge of web services, XML-based protocols (SOAP, WSDL, UDDI), and distributed application development. The course also emphasizes building Windows-based applications, working with assemblies, memory management, and integrating components such as COM and remoting services, thereby equipping students with the skills required to design, develop, and deploy modern enterprise-level applications using the .NET platform.

MJE-02T: .NET Framework(Theory) Credits 03

Module -I: Introduction to .NET Framework (08 Lectures)
An Overview of NET, Defining NET, Web Services, The NET Framework,The Common Language Runtime, CLR Based Languages, The NET Framework Class Library, The .NET Compact Framework, NET My Services, The NET Enterprise Servers, A NET Scenario.

Module -II: Web Services and Protocols (06 Lectures)
Web Services, Describing Web Services, Access to Internet Applications, B2B Integration, A Web Services Scenario, XML, WSDL, SOAP, UDDI, Future Directions for Web Services.

Module -III: Architectural Overview of .NET Framework (10 Lectures)
The Common Language Runtime, The Common Type System, Introducing the Common Type System, The Common Language Specification, Compiling Managed Code, Microsoft Intermediate Language, Metadata, Manifests, Categorizing Assemblies, Loading Assemblies, Compiling MSIL, Securing Assemblies, Garbage Collection, Application Domains.

Module -IV: Application Development using Several Languages in .NET (16 Lectures)
.NET Languages, VB and C#, NET Languages, Overview of the NET Framework, The System Namespace, A Survey of Systems Subordinate, System , System Runtime Serialization, System XML, The XML Technology Family, What System XML Provides, System Reflection, System Runtime Remoting, An Overview of the Remoting Process, Choosing a Channel, Creating and Destroying Remote Object, System Enterprise Services, Accessing COM Objects, Accessing Non COM DLLs, Building GUIs Using Windows Forms, Windows Forms Controls.

Reference Books:

1. Understanding .NET: a tutorial and analysis By David Chappell
2. Beginning ASP.NET 2.0 with C#, Author:Hart Chris

3. C# and the .NET platform, Author: Troelsen Andrew
4. Core C# and .NET, Author:Perry Stephen C

MJE-02P: .NET Framework (Practical) **Credits 01**

1. Write a C# program to demonstrate the basic structure of a .NET console application and print "Hello .NET".
2. Write a VB program to display the initials of your name and surname using .NET.
3. Create a program to demonstrate the use of different data types in .NET.
4. Write a program to show the difference between value types and reference types.
5. Develop a simple application to demonstrate exception handling in .NET.
6. Create a program to illustrate the use of the .NET Framework Class Library (e.g., using System.Math, System.DateTime).
7. Create a simple XML file and write a C# program to read and display its contents.
8. Write a program to parse XML using XmlReader or XmlDocument.
9. Write a program to demonstrate the use of namespaces and assemblies in .NET.
10. Create a program to illustrate boxing and unboxing using the Common Type System.
11. Write a program to generate and inspect metadata of an assembly using reflection.
12. Create a multi-file assembly and demonstrate how it is compiled and executed.
13. Develop a Windows Forms application to perform basic arithmetic operations.
14. Create a GUI application using Windows Forms with different controls (Button, TextBox, Label).
15. Write a C#.NET program demonstrating file handling (read/write text files).
16. Write a VB.NET program to create a Web Browser of your own and show the recent surfing history of users.
17. Create a Desktop application to perform Tic-Tac-Toe game with both User vs User and User vs System mode.
18. Create a program to access a COM component or external DLL from a .NET application.
19. Make a web application for calendar based events with notifications.
20. Develop a simple client-server application using .NET Remoting.

Minor

UG/V/COMP/3/MI-C5: Operating System Credits 04

Objective of the course

- Provide a comprehensive understanding of fundamental concepts and functions of modern operating systems.
- Cover the architecture and components of operating systems, including process management, memory management, file systems, and input/output systems.
- Teach about concurrency, process synchronization, and inter-process communication for efficient task management.

- Emphasize the role of operating systems in resource allocation and system security. Offer hands-on experience in implementing and configuring operating system features through labs and projects.
- Explore various operating systems like Windows, Linux, and macOS to understand their differences and commonalities.
- Delve into virtualization and distributed systems to highlight current trends in operating system design.
- Equip students with the skills to analyze, design, and optimize operating systems, preparing them for advanced study and careers in systems programming and software engineering.

MI-C5T: Operating System(Theory) Credits 03

Course contents:

Module I: Introduction (10 Hrs.)
Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems , time sharing systems; operating systems for personal computers & workstations, process control & real time systems. Case study on Linux system. Cloud computing. Linux evolution and Linux distros. Linux file system.

Module II: Operating System Organization (6 Hrs.)
Processor and user modes, kernels, system calls and system programs.

Module III: Process Management (16 Hrs.)
System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent processes, critical section, semaphores, methods for inter-process communication; deadlocks.

Module IV: Memory Management (12 Hrs.)
Physical and virtual address space; memory allocation strategies – fixed and variable partitions, paging, segmentation, virtual memory Module V: I/O Management Directory structure, file operations, file allocation methods, device management.

Recommended Books:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.

2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.

3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.

4. W. Stallings, Operating Systems, Internals & Design Principles , 5th Edition, Prentice Hall of India. 2008.

5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

MI-C5P: Operating System Lab (Practical) **Credits 01**

1. Implement a simple fork() program.
2. Implement number of times hello is printed.
3. Implement a program where parent and child execute.
4. Implement a FCFS program where process arrival times are same.
5. Implement a FCFS program where process arrival times are different.
6. Write program to implement Round Robin scheduling algorithm.
7. Write program to implement SJF scheduling algorithm.

SEMESTER-VI

UG/VI/COMP/3/MJ-B4 Same as A4 **Credits 04**

UG/VI/COMP/3/MJ-B5 Same as A5 **Credits 04**

UG/VI/COMP/3/MJ-B6 Same as A6 **Credits 04**

UG/VI/COMP/3/MJE-02: Machine Learning **Credits 04**

Objective of the course

- Learn the basics of machine learning, understanding its uses, challenges, and various applications.
- Build practical data skills, covering data collection, analysis, visualization, and preparation.
- Become skilled in using classification and regression algorithms, including selecting, training, and evaluating models.
- Dive into advanced clustering and specialized applications, using methods like K-Means, DBSCAN, and others.

Course content

Module 1: Fundamentals of Machine Learning; **(14 Hrs)**

Introduction to Machine Learning: What is Machine Learning? Why Use Machine Learning? Types of Machine Learning Systems, Main Challenges of Machine Learning, Applications of Machine Learning. Why Python, scikit-learn, Essential Libraries and Tools.

Module 2: Data Preparation: **(12 Hrs)**

Working with Real Data, look at the Big Picture, Get the Data, Discover and Visualize the Data to Gain Insights, Prepare the Data for Machine Learning Algorithms, Select and Train a Model.

Module 3: Supervised Learning: (12 Hrs)

Classification and Regression, Some Sample Datasets, k-Nearest Neighbours, Linear Models, Naive Bayes Classifiers, Decision Trees.

Module 4: Unsupervised Learning: (12 Hrs)

Clustering, K-Means, Limits of K-Means, using clustering for image segmentation, Using Clustering for Preprocessing, Using Clustering for Semi-Supervised Learning, DBSCAN, Other Clustering Algorithms.

Recommended Books:

1. Andreas . C. Müller and S. Guido, "Introduction to Machine Learning with Python," O'Reilly, 2017.
2. Amanda . Casari and Alice . Zheng, "Feature Engineering for Machine Learning," O'Reilly Media, Inc., 2018, p. 218.
3. A. Géron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow," O'Reilly Media, Inc., 2022.
4. Ian Goodfellow, Yoshua. Bengio, and Aaron. Courville, "Deep Learning," MIT Press, 2016.
5. S. Rashka and V. Mirdzhalili, "Machine Learning and Deep Learning with Python, scikitlearn, and TensorFlow 2," Packt, Birmingham and Mumbai, 2020.
6. S. Shalev-Shwartz and S. Ben-David, "Understanding Machine Learning: From Theory to Algorithms," Cambridge University Press, 2014.

MJE-02: Machine Learning Lab Credit 01

1. Install and set up Python and essential libraries like NumPy and pandas.
2. Introduce scikit-learn as a machine learning library.
3. Install and set up scikit-learn and other necessary tools.
4. Write a program to Load and explore the dataset of .CVS and excel files using pandas.
5. Write a program to Visualize the dataset to gain insights using Matplotlib or Seaborn by plotting scatter plots, bar charts.
6. Write a program to Handle missing data, encode categorical variables, and perform feature scaling.
7. Write a program to implement a k-Nearest Neighbours (k-NN) classifier using scikit- learn and Train the classifier on the dataset and evaluate its performance.
8. Write a program to implement a linear regression model for regression tasks and Train the model on a dataset with continuous target variables.
9. Write a program to implement a decision tree classifier using scikit-learn and visualize the decision tree and understand its splits.
10. Write a program to Implement K-Means clustering and Visualize clusters.

Minor

UG/VI/COMP/3/MI-C6: Database Management System **Credits 04**

Module- I: Introduction **(6 Hrs.)**
Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Data Abstraction, Three Schema architecture of DBMS.

Module- II: Entity Relationship (ER) Model **(6 Hrs.)**
Entity Set, Simple and composite Attribute, Single valued and multivalued attribute, Relationship sets, Mapping cardinality, keys, , Entity Relationship Diagram : Need for E-R Model, Various steps of database design, Mapping Constraints, E-R diagram, Subclass, Generalization, Specialization, Aggregation, Strong Entity-Weak Entity.

Module- III: Relational Algebra **(10 Hrs.)**
Select operation, Project Operation, Set operations (union, intersection, difference), Join operations, Division operation, outer join and outer union, Examples queries in Relational Algebra.

Module- IV: SQL **(8 Hrs.)**
Concept of DDL, DML. Basic Structure Relational databases and tables, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Sub queries.

Module- V: Relational Model and Relational Database Design **(10 Hrs.)**
Concept of Relational Model, Design Issues, Keys, Closure set, Functional Dependency, Different anomalies in designing a Database., Normalization using functional dependencies, Decomposition, 3NF, Boyce-Codd Normal Form.

Module- VI: Transaction Management **(10 Hrs.)**
ACID properties, Transaction definition, properties, transaction state diagram, commit and rollback, Serializability (Conflict and View), Concurrency control, lock based protocols, Two phase locking, Timestamp ordering protocol, Recovery management, Deadlock handling and prevention.

References:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, DatabaseSystem Concepts 6th Edition, McGraw
Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

MJ-A6P: Database Management Systems (Practical) **Credits 01**

2. An inventory database has the following tables:

ITEM (Item_Code, Item_Name, Price)

PURCHASE (Item_Code, Quantity, Purchase_date)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

f. Create the tables with the above attributes and enter 5-7 tuples into each table.
g. Display the List of all items with their price which have minimum 10 quantity.
h. Display the List of items which are not purchased by anyone.
i. Display all the item with their Purchase date.
j. Display all items according to their price.

2. Create the following tables with the fields given below:

TEACHER (T_ID, Department, Year of Exp, Name)

SUBJECT (Sub_Paper_ID, T_ID, Title_of Paper, Programme, Semester)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

f. Create the tables with the above attributes and enter 5-7 tuples into each table.
g. Display Name and Year-Of-Exp of all the teachers of "Computer Science" department.
h. List the Subject Paper which are handled by T_ID = 101.
i. List the name of Programme and Semester of paper titled "C Programming".
j. List the name of teacher who are allotted for 4th semester.

3. Create the following tables with the fields given below:

PRODUCT (Product_ID, Supplier_ID, Proudct_Name, Category, Price, Quantity)

SUPPLIER (Supplier_ID, S_Name, Area, City)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

f. Create the tables with the above attributes and enter 5-7 tuples into each table.
g. List price and product name of all the products whose Quantity is more than 100.
h. Display product name and product ID of the products which are supplied by Supplier_iD = 10111.
i. Display the number of products which price is more than Rs. 1,500 and Quantity is more than 5.
j. Display name and area of all the suppliers of city "Kolkata".

4. Create the following tables with the fields given below:

DOCTOR(Did, DName, Dept_name, Salary, Joining_date)

PATIENT(Pid, Did, Pname, P_address, Phone_no, Admitted_dept)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

f. Create the tables with the above attributes and enter 5-7 tuples into each table.
g. Find the list of patients admitted in “Orthopedic” department.
h. Display the name, Department and Salary of doctor who get maximum salary.
i. Display the total number of patients admitted under “Dr. M Chatterjee”.
j. List the number of patients in each Department.

5. Create the following tables with the fields given below:

STUDENT (Student ID, Name, Programme, Teacher ID)

TEACHER (Teacher ID, Department, Name, City, Specialization)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

f. Create the tables with the above attributes and enter 5-7 tuples into each table.
g. Display total number of students registered in "MCA" programme.
h. Display name of all the students who are associated with TeacherID = 123.
i. Display name, specialization and city of all the teachers who are associated with "Compute Science" department.
j. Display unique Department from the Teacher Table.

6. Create the following tables with the fields given below:

Depertment (Dept_no, D_name, loc)

Employee (Empno, E_name, job, mgr, hiredate, sal, comm, Dept_no)

Select appropriate e primary keys and foreign keys. Select appropriate data types for all the fields.

f. Create the tables with the above attributes and enter 5-7 tuples into each table.
g. Update the employee salary by 15%, whose experience is greater than 10 years.
h. Display the manager who is having maximum number of employees working under him.
i. Display Name of all the employees where the third letter of their name is _A‘.
j. Display Unique Listing of all Jobs that are in Department no 30.