



BON SECOURS COLLEGE FOR WOMEN (AUTONOMOUS)

Accredited with 'A++' Grade by NAAC in Cycle II
Affiliated to Bharathidasan University, Tiruchirappalli
UGC Recognized 2(f) and 12(B) Institution
Vilar Bypass, Thanjavur – 613 006

DEPARTMENT OF COMPUTER SCIENCE

B. Sc., Computer Science

SEMESTER-I

CORE COURSE - I

PRINCIPLES OF PROGRAMMING WITH C

COURSE CODE: 24UCP401

Contact Hours: 4

Credits: 4

Internal Marks: 25

External Marks: 75

SYLLABUS

UNIT I : Principles of Computer

Principles of Computers: Introduction to computer –Classification of Computers-Basic Anatomy of a Computer System-Input Devices- -Output Devices-Memory Management –Programming Languages- Algorithms – Flow charts – Developing algorithms and flowcharts for solving simple problems using sequential, selection and iterative programming Structures.

UNIT II: Overview and Decision Making, Looping

History of C and its importance – Structure of a C program – Data Types – Constants and Variables – Operators and Expressions – Control structures – Looping structures.

UNIT III: Arrays & Functions

Arrays – single dimensional arrays – declaration –memory representation– initialization and access. 2D arrays and multidimensional arrays- Strings – defining strings, initializing, accessing, character handling functions- Functions – definition-declaration-prototypes and function call- actual and formal arguments-types of functions- call by value-call by reference-nesting of functions-recursive functions

UNIT – IV: Pointers

Introduction – Pointer Expressions – Chain of Pointers –Pointers and Arrays – Array of Pointers – Pointers as function arguments – Functions returning Pointers – Pointers to Functions – Function pointer – Pointers and Structures.

UNIT V: Structure & Files

Structures: Introduction – Defining a structure – Declaration of structure – Accessing Structures members – Array of Structures – Structures within structures – Structures and functions – Structures and Pointers – Union. Files: Opening and closing files – Operations on files.

Text Books:

1. E. Balagurusamy, “Programming in ANSI C”, McGraw Hill, 8th Edition, 1st January 2024.

Reference Books:

1. JyotiP Mirani &Vikas B. Agarwal, “Computer Fundamentals”, 2nd Edition, Nirali Prakashan, 2019.
2. Yashavant Kanetkar, “Let Us C”, 18th Edition, BPB Publications, 2021.
3. Szuhay, Jeff, and Szuhay, Jeff, “Learn C Programming: A Beginner's Guide to Learning C Programming the Easy and Disciplined Way”, Packt Publishing, 2020.
4. Jena, Sisir Kumar, and Jena, Sisir Kumar, “C Programming: Learn to Code”, 1st Edition, CRC Press, 2021.
5. Reema Thareja , “Programming in C”, 3rd Edition, Oxford University Press, 2023.

E-Resources:

1. <https://www.w3resource.com/c-programming-exercises/>
2. <https://www.javatpoint.com/c-programming-language-tutorial>
3. <https://www.geeksforgeeks.org/best-c-programming-courses-for-beginners/>
4. <https://www.tutorialspoint.com/cprogramming/index.htm>
5. https://onlinecourses.nptel.ac.in/noc22_cs40/preview

CORE COURSE - I
DIGITAL COMPUTER FUNDAMENTALS
COURSE CODE: 24UCP402

Contact Hours: 4
Credits: 4

Internal Marks: 25
External Marks: 75

SYLLABUS

UNIT 1: Number Systems

Digital Systems-Binary numbers- Number base conversions-Octal, Hexadecimal -one's complement and 10's complement, Binary Arithmetic, Fixed point representation and Floating-point representation of-Signal -Binary Numbers-Binary Codes.

UNIT II: Boolean Algebra & Logic Gates

Boolean Algebra-Basic definitions Axiomatic Definition of Boolean Algebra -Basic Theorems and Properties of Boolean Algebra-Boolean Functions- Canonical and standard forms -The AND gate, The OR The NAND gate, NOR gate, The Exclusive-OR gate and Exclusive-NOR gate- Boolean Algebra and Logic Simplification DE Morgan's Theorems, Boolean Expressions and Truth Tables, The Karnaugh Map - SOP, POS, Don't care conditions.

UNIT-III: Combinational Logic Circuits

Basic combinational Logic Circuits, Implementing Combinational Logic. The Universal Property of NAND and NOR Gates. Functions of Combinational Logic Adders, Subtractors, Decoders, Encoders, Code Converters, Multiplexers, Demultiplexers

UNIT-IV: Sequential Circuit Design

Flip flops (RS, Clocked RS, D, JK, JK Master Slave, T, Counters, Shift registers and their types, Counters: Synchronous and Asynchronous counters

UNIT-V: Memory and Programmable Logic

Introduction, Random access memory, memory decoding, error detection and correction, read only memory, Flash Memory Special Types of Memories, Magnetic and Optical Storage programmable logic array, sequential programmable devices.

Text Book :

1. M. Morris Mano, Michael D. Ciletti, "Digital Design", 6th Edition, Pearson Education, 2018.

Reference Book :

1. Donald P leach, Albert Paul Malvino, GoutamSaha, Digital Principles and Application, 8th Edition., McGraw Hill education (India) Private Limited, 2015.
2. Prof. Satish Jain Shashi Singh, "Digital Computer Fundamentals", 1st Edition, BPB Publications, 2017.
3. Ronald Tocci, Neal Widmer , Gregory Moss," Digital Electronics Principles and applications, 12th Edition, Pearson, 2017.
4. Digital Electronics Circuits and Systems, BishnuCharanSarkar and Suvra Sarkar, 1st Edition, Sad-UI-Tan Prakasani, Burdwan, 2018.

E-Resources:

1. <http://nptel.ac.in/courses.php>
2. <http://jntuk-coeerd.in>
3. <http://electrical4u.com/>

CORE LABORTORY - I
PRINCIPLES OF PROGRAMMING WITH CLAB
COURSE CODE: 24UCP451

Contact Hours: 4
Credits: 2

Internal Marks: 40
External Marks: 60

SYLLABUS

1. Write C program
 - (i) To convert temperature from degree Centigrade to Fahrenheit,
 - (ii) Find whether given number is even or odd,
 - (iii) Find the greatest of three numbers.
2. Write a C program to find the sum, average, standard deviation for a given set of numbers.
3. Write a C program to print magic square of order n where $n > 3$ and n is odd.
4. Write C program to solve quadratic equation using functions.
5. Write C program
 - (i) To swap two numbers using call by value
 - (ii) To change the elements of an array using call by reference
6. Write C program to count the number of vowels in the given sentence.
7. Write C program to find the factorial of a given number using a recursive function.
8. Write C program to reverse a number using pointer.
9. Write C program to check whether the given string is a palindrome using pointers.
10. Write C program to print the students mark sheet assuming roll no, name, and marks in 5 subjects in a structure. Create an array of structures and print the mark sheet in the university pattern.
11. Write a function using pointers to add two matrices and to return the resultant matrix to the calling Function.
12. Write a C program which receives two file names as arguments and check whether the file contents are same or not. If same delete the second file.

SEMESTER-II

CORE COURSE – III DATABASE MANAGEMENT SYSTEMS COURSE CODE: 24UCS501

Contact Hours: 4

Credits: 4

Internal Marks: 25

External Marks: 75

UNIT I - INTRODUCTION

Introduction: Database-System Applications- Purpose of Database Systems - View of Data -Database Languages - Relational Databases - Database Design -Data Storage and Querying Transaction Management -Data Mining and Analysis - Database Architecture - Database Users and Administrators - History of Database Systems.

UNIT II – RELATIONAL MODEL

Relational Model: Structure of Relational Databases -Database Schema - Keys – Schema Diagrams - Relational Query Languages - Relational Operations Fundamental Relational Algebra Operations Additional Relational-Algebra Operations- Extended Relational Algebra Operations - Null Values - Modification of the Database.

UNIT III – SQL

SQL Overview of the SQL Query - Language - SQL Data Definition - Basic Structure of SQL Queries - Additional Basic Operations - Set Operations - Null Values Aggregate Functions - Nested Subqueries - Modification of the Database -Join Expressions - Views - Transactions - Integrity Constraints - SQL Data Types and Schemas – Authorization.

UNIT IV –RELATIONAL LANGUAGE

Relational Languages: The Tuple Relational Calculus - The Domain Relational Calculus Database Design and the E-R Model: Overview of the Design Process - The Entity Relationship Model - Reduction to Relational Schemas - Entity-Relationship Design Issues - Extended E-R Features - Alternative Notations for Modeling Data - Other Aspects of Database Design.

UNIT V –RELATIONAL DATABASE

Relational Database Design: Features of Good Relational Designs - Atomic Domains and First Normal Form - Decomposition Using Functional Dependencies – Functional Dependency Theory - Decomposition Using Functional Dependencies - Decomposition Using Multivalued Dependencies-More Normal Forms - Database-Design Process

Text Book:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, *"Database System Concepts"*, McGraw-Hill Education, 7th Edition, 2019.

Reference Books:

1. Ramez Elmasri and Shamkant B. Navathe, *"Fundamentals of Database Systems"*, Pearson Education, 7th Edition, 2017.
2. Raghu Ramakrishnan and Johannes Gehrke, *"Database Management Systems"*, McGraw-Hill Education, 3rd Edition, 2003.
3. Thomas Connolly and Carolyn Begg, *"Database Systems: A Practical Approach to Design, Implementation and Management"*, Pearson Education, 6th Edition, 2014.
4. C. J. Date, *"An Introduction to Database Systems"*, Pearson Education, 8th Edition, 2004.
5. Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom, *"Database Systems: The Complete Book"*,

Pearson Education, 2nd Edition, 2008.

E-resources:

1. <https://www.geeksforgeeks.org/dbms/>
2. <https://www.javatpoint.com/dbms-tutorial>
3. <https://www.tutorialspoint.com/dbms/index.htm>
4. <https://www.w3schools.com/sql/>

CORE LABORTORY - II
DATABASE MANAGEMENT SYSTEMS LAB
COURSE CODE: 24UCS551

Contact Hours: 4

Credits: 2

Internal Marks: 40

External Marks: 60

SYLLABUS

- 1. Create a table and perform the following basic mysql operations**
 - a. Set the primary key
 - b. Alter the structure of the table
 - c. Insert values
 - d. Delete values based on constraints
 - e. Display values using various forms of select clause
 - f. Drop the table
- 2. Develop mysql queries to implement the following set operations**
 - a. Union
 - b. Union all
 - c. Intersect
 - d. Inter sectall
- 3. Develop mysql queries to implement the following aggregate functions**
 - a. Sum
 - b. Count
 - c. Average
 - d. Maximum
 - e. Minimum
 - f. Group by clause & having clause
- 4. Develop mysql queries to implement following join operations**
 - a. Natural join
 - b. Inner join
 - c. Outer join-left outer, right outer, full outer
 - d. Using join conditions
- 5. Develop mysql queries to implement nested sub queries**
 - a. Set membership(int, notint)
 - b. Set comparison(some, all)
 - c. Empty relation(exists, notexists)
 - d. Check for existence of Duplicate tuples(unique,notunique)
- 6. Develop mysql queries to create views and expand it.**
- 7. Develop mysql queries to implement**
 - a. String operations using%
 - b. String operations using
 - c. Sort the element using asc, desc

[*create necessary reletions with requires attribute]
- 8. Consider the following database for a banking enterprise** BRANCH(branch-name: string, branch-city: string, assets: real) ACCOUNT(accno: int, branch- name: string, balance: real) DEPOSITOR(customer- name: string, accno: int) CUSTOMER(customer-name:string,customer-street:string,customer-city: string) LOAN(loan-number: int, branch-name: string, amount: real) BORROWER(customer- name: string, loan- number: int)
 - i. Create the above tables by properly specifying the primary keys and the
 - ii. Foreign keys
 - iii. Enter atleast five tuples for each relation
 - iv. Find all the customers who have atleast two accounts at the Main Branch.

- v. Find all the customers who have an account at all the branches located
 - vi. In a specific city.
- Demonstrate how you delete all account tuples at every branch located in a specific city.
- Generate suitable reports.

Create a suitable frontend for querying and displaying the results

9. Develop mysql queries to create a views and expand it.

10. Develop mysql queries to implement

- d. String operations using%
 - e. String operations using__‘
 - f. Sort the element using asc, desc
- [*create necessary relations with requires attribute]

Consider the following database for a banking enterprise BRANCH(branch-name:string, branch-city: string,assets:real) ACCOUNT(accno:int, branch-name:string, balance:real) DEPOSITOR(customer-name:string, accno:int) CUSTOMER(customer-name:string,customer-street:string,customer-city: string) LOAN(loan-number:int,branch-name:string,amount:real) BORROWER(customer-name:string, loan-number:int)

- vii. Create the above tables by properly specifying the primary keys and the
 - viii. Foreign keys
 - ix. Enter atleast five tuples for each relation
 - x. Find all the customers who have atleast two accounts at the Main Branch.
 - xi. Find all the customers who have an account at all the branches located
 - xii. In a specific city.
 - xiii. Demonstrate how you delete all account tuples at every branch located in a specific city.
- Generate suitable reports.
- xiv. Create a suitable frontend for querying and displaying the results.

Tools Required

- Microsoft SQL Server
- Oracle SQL Developer

E-Resources

- [Database Management System Lab Component\(BCS403\) My Blogosphere](#)
- <https://code.org/educate/weblab>
- <https://www.freecodecamp.org/https://designlab.wisc.edu/resources/>
- <https://webflow.com/resources>

SEMESTER-III

24UCP403: Python Programming Core Course – IV

Credits: 4
Contact Hours: 4

Internal Marks: 25
External Marks: 75

Syllabus

Unit I

Introducing Python - Python on Different Operating Systems: Python on Windows- Python on macOS - Python on Linux -Variables and Assignment Statements – Arithmetic – Function print and an Introduction to Strings.

Unit II

Decision Making: The if Statement and Comparison Operators - Objects and Dynamic Typing-Control Statements: If Statement-If Else and If Elif Else Statements -While Statement, For Statement -Iterables, Lists and Iterators -Builtin Range Function -Augmented Assignments –Sequences: Lists and Tuples-Dictionaries and Set-controlled Iteration; Formatted Strings –Sentinel controlled Iteration - Builtin Function Range-Break and Continue Statements.

Unit III

Functions: Defining Functions-Functions with Multiple Parameters- Python Standard Library -math Module Functions -Using IPython Tab Completion for Discovery - Default Parameter Values -Keyword Arguments Arbitrary Argument Lists-Methods: Functions That Belong to Objects- Scope rules- Recursion- Files: Text File Processing - Updating Text Files - Handling Exceptions- finally Clause -Explicitly Raising an Exception.

Unit IV

Object Orientation: Custom Class Account- Definition-Composition: Object References as Members of Classes Controlling Access to Attributes-Properties for Data Access- Account Class Definition-Composition: Object References as Members of Classes -Controlling Access to Attributes-Properties for Data Access- Duck Typing and Polymorphism- Operator Overloading- Namespaces and Scopes.

Unit V

NumPy Basics: Arrays and Vectorized Computation: The NumPy ndarray: A Multidimensional Array Object- Pseudorandom Number Generation- Universal Functions: Fast Element-Wise Array Functions- Array-Oriented Programming with Arrays- File Input and Output with Arrays- Linear Algebra- Getting Started with pandas: Introduction to pandas Data Structures- Essential Functionality.

Text Book:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021(Unit-I-IV)
2. Python for Data Analysis Data Wrangling with pandas, NumPy, and Jupyter, Wes McKinney, oreilly ,3rd Edition,2022(Unit-V)

Reference Books:

1. Python Crash Course by Eric Matthes, 3rd Edition, No Starch Press, January 2023
2. Martin C. Brown, Python: The Complete Reference, McGraw Hill Education (India), 2024.
3. Jeeva Jose and P. SojanLal, “Introduction to Computing and Problem Solving with PYTHON”, Khanna Book Publishing Co, 2016.
4. Phill Vega., "Python Programming 2020: Guide to Learn Python, Step by Step, with Practical Exercises, Tips and Tricks", Independently Published, 2020.
5. Reema Thareja,"Python Programming: Using Problem Solving Approach", Oxford University Press, 2019.
6. Kanetkar Aditya Yashavant, Kanetkar, Let Us Python, 2nd Edition, BPB Publications, 2020.

E-Resources:

- <https://www.tutorialspoint.com/python/index.htm>
- <https://www.w3schools.com/python/default.asp>
- <https://www.learnpython.org/>
- <https://nptel.ac.in/courses/106106145>
- <https://www.tpointtech.com/python-tutorial>

Core Laboratory – III
24UCP452: Python Programming Lab

Credits: 2
Contact Hours: 4

Internal Marks: 40
External Marks: 60

Syllabus

1. Write a Python program to simulate an ATM using conditional statements.
2. Write a Python program for a BMI calculator with health condition output.
3. Write a Python program to count word frequency in a sentence.
4. Write a Python program to compute the factorial of a number using both loops and recursion.
5. Write a Python program for an inventory management system using dictionary.
6. Write a Python program for an employee attendance tracker using dictionary.
7. Write a Python program to add two matrices of the same size and display the result.
8. Write a Python program to find the transpose of a matrix.
9. Write a Python program for a bank transaction system using functions.
10. Write a Python program for a to-do list application using file handling.
11. Create a program to count the number of times a specific word appears in a given text file.
12. Implement a BankAccount class with methods to deposit, withdraw, and display the balance.
13. Create a Vehicle class and derive Car and Bike subclasses from it.
14. Write a Python program Basic Array Operations using NumPy
15. Write a Python program to create and display a DataFrame using Pandas.
16. Write a Python program to implement a Stack and queue data structure using a list. Include push, pop, peek, and is_empty operations.
17. Write a Python program to implement a singly linked list with the following operations:
Insertion at the end Deletion of a node by value Display the list
18. Write a Python program to implement the following sorting algorithm using Bubble Sort.
19. Write a Python program to implement Linear Search and Binary Search.
20. Write a Python program to implement graph traversal using Breadth-First Search (BFS) and Depth-First Search (DFS).

Tools Required

- Python Interpreter
- Integrated Development Environments (IDEs) / Code Editors
- Command Line / Terminal
- Package Manager.
- Additional Python Libraries.
- IPython / Jupyter Notebook.
- Text Editors.
- Version Control System.

E-Resources

- <https://docs.python.org/3/>
- <https://www.w3schools.com/python/>
- <https://www.geeksforgeeks.org/python-programming-language/>

Core Course - V
24UCP404: Data Structures and Algorithms

Credits: 4
Contact Hours: 4

Internal Marks: 25
External Marks: 75

Syllabus

Unit I

Introduction of Algorithms, Analysing Algorithms: Theoretical Foundation, Algorithms and its Specification, Random Access Machine Model, Counting Primitive Operations, Notion of best case, average case and worst case, Characterizing Run Time, Use of asymptotic notation, Big-Oh Notation, Little-Oh, Omega and Theta Notations.

UNIT II

Arrays: Sparse Matrices–Representation of Arrays- Stacks and Queues - Fundamentals–Evaluation of Expression Infix to Postfix Conversion - Multiple Stacks and Queues - Linked List: Singly Linked List - Linked Stacks and Queues – Polynomial Addition- More on Linked Lists – Sparse Matrices – Doubly Linked List and Dynamic – Storage Management –Garbage Collection and Compaction.

UNIT III

Basic Terminology – Binary Trees – Binary Tree Representations – Binary Trees-Traversal-More on Binary Trees – Threaded Binary Trees – Binary Tree.Representation of Trees – Counting Binary Trees. Graphs: Terminology and Representations-Traversals, Connected Components and Spanning Trees, Shortest Paths and Transitive Closure.

UNIT – IV

Introduction to Algorithms - Priority Queues - Heaps – Heap Sort – Divide and Conquer Algorithms - Binary Search -Merge Sort – Quick Sort — Finding the Maximum and Minimum.

UNIT - V

Greedy Method: The General Method – Knapsack Problem – Minimum Cost Spanning Trees– Optimal Merge Patterns- Single-Source Shortest Paths. Back Tracking: The General Method – The 8-Queens Problem – Sum of Subsets – Graph Coloring – Hamiltonian Cycles.

Text Book(s):

1. Ellis Horowitz, Sartaj Sahni, "Fundamentals of Data Structure", Galgotia Publications (UNIT I-III)
2. Ellis Horowitz, Sartaj Sahni and Sangu Thevar Rajasekaran, "Computer Algorithms", University Press. (UNIT IV&V)

Reference Book(s):

1. Seymour Lipschutz, "Data Structures", Tata McGraw Hill, Schaum's Outline Series, 2014.
2. Micheal T. Goodrich and Roberto Tamassia: Algorithm Design: Foundations,
3. Analysis and Internet examples (John Wiley & Sons, Inc., 2006).
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein,
5. Introduction to Algorithms, 3rd Edition, McGraw-Hill, 2010.

E-Resources:

- <https://www.geeksforgeeks.org/dsa-tutorial-learn-data-structures-and-algorithms/>
- https://www.google.com/search?q=HackerRank&rlz=1C1ONGR_enIN961IN962&oq=HackerRank&aqs=chrome..69i57.1016j0j4&sourceid=chrome&ie=UTF-8

SEMESTER-IV

Core Course V

24UCS502: Introduction to J2EE

Credits: 4

Contact Hours: 4

Internal Marks: 25

External Marks: 75

Syllabus

UNIT I

Introduction: Object Oriented concepts - History of Java - Java buzzwords - JVM architecture - Data types - Variables - Scope and life time of variables - Arrays - Operators - control statements - type conversion and casting - simple java program - constructors - methods - Static block - Static Data - Static Method String and String Buffer Classes.

UNIT II

Inheritance: Basic concepts - Types of inheritance - Member access rules - Usage of this and Super key word – Abstract classes - Dynamic method - Usage of final keyword- Polymorphism-Inherit from Different Classes – Interface: Basics – Multiple Interfaces – Multiple Inheritance using Interfaces- Packages: User Defined Package.

UNIT III

Exception Handling: try – catch - throw - throws – finally – Built-in exceptions - Creating own Exception classes - User Defined Exception – Functions - User Defined Functions : Definition of Functions- Syntax of User Defined Methods- Types of Methods Based on Parameters and Return Type- Method Overloading- Calling Methods- Recursion- Multithreaded Programming: Thread Class - Runnable interface – Synchronization – Using synchronized methods – Using synchronized statement – Inter thread Communication – Deadlock – File Handling - Integrated Development Environment (IDE).

UNIT IV

RMI Introduction, RMI Architecture, Client Server Application using RMI - Servlet Introduction, Servlet Life Cycle(SLC), Types of Servlet- Request and Response Handling- Servlet and Sessions- Cookies Servlet Configuration with Deployment Descriptor, Working with Servlet Context and Servlet Config Object, Attributes in Servlet – Servlet Security- Deploying Servlets.

UNIT V

Introduction to JSP - JSP Syntax – JSP Life Cycle - JSP Implicit Objects Comparison with Servlet, JSP Architecture - JSP Standard Tag Library (JSTL)- JSP and JavaBeans- JSP Expression Language- Form Handling in JSP- Session Management in JSP- MVC Architecture in JSP- Error Handling and Debugging in JSP- JDBC: Work with JSP.

Text Book

1. Herbert Schildt "Java: The Complete Reference" McGraw-Hill 11th Edition (2021)

Reference book

2. Cay S. Horstmann "Core Java Volume II - Advanced Features" Pearson 11th Edition (2021)
3. Herbert Schildt "Java Programming: From the Beginning to the Advanced Level" McGraw-Hill Education Edition (2020)

E- Resources

1. <https://archive.org/details/j2eecomplete0000keog>
2. <https://link.springer.com/book/10.1007/978-1-4471-0017-1>

CORE LABORATORY - IV

24UCS552: J2EE LAB

Contact Hours: 4

Credits: 2

Internal Marks: 40

External Marks: 60

Syllabus

1. Write a java program to find all even and odd integers up to a given integer.
2. Write a java program that checks whether a given string is palindrome or not Program.
3. Write a java program that displays the number of characters, lines, words, white spaces in a text file.
4. Write a java program to represent Abstract class.
5. Write a java program to create user defined package.
6. Write a Java program to demonstrate the concept of exception handling using the `try` and `catch` blocks.
7. Write a Java Program to Demonstrate throw and throws.
- 8 . Write a Java program to read a large file efficiently using Buffered Reader and process each line.
9. Write a Java Program to Write Text to a File Using Buffered Writer.
10. Write a Java Program to Read from a File Using File Reader.
11. Write a Java Program to Write to a File Using File Writer.
12. Write a Java Program to Demonstrate Data Input Stream
13. Write a Java Program to Demonstrate Data Output Stream.
13. Write a Java Program to Develop a Remote Method Invocation.
14. Create a simple Java Servlet that accepts user input (name) through an HTML form and displays a greeting message using that input.
15. Write a Servlet Program to implement Session Tracking.
16. Write a Servlet Program to Check Authentication for Users Using Cookies.
17. Write a Servlet Program and use JDBC in it.
18. Write a Simple Program for JSP.
19. Write a JSP Program that works with JDBC.
20. Create a Simple Java bean that represents a user with properties like name, age and Email. Implement Getter and Setter Methods.

SEMESTER-V

Core Course – VII: PRINCIPLES OF OPERATING SYSTEM

Course Code: 24UCS503

Credits: 4

Contact Hours/Week: 4

Internal marks: 25

External marks: 75

Syllabus

Unit I - Introduction of Operating System

Definition of Operating System – Mainframe Systems – Desktop Systems Multiprocessor Systems – Distributed Systems – Clustered Systems – Real-Time Systems – Handheld Systems – Operating System Structure – System Components – Operating System Services – System Calls – System Programs – System Design and Implementation

Unit II - Process Management

Process Concepts – Process Scheduling – Operations on Processes – Cooperating Processes – Inter-Process Communication – CPU Scheduling – Scheduling Criteria – Scheduling Algorithms – Multiprocessor Scheduling – Real-Time Scheduling

Unit III - Process Synchronization

Critical Section Problem – Synchronization Hardware – Semaphores – Classical Problems of Synchronization – Critical Regions – Monitors – Deadlock Characterization – Handling Deadlocks – Deadlock Prevention – Deadlock Avoidance – Banker's Algorithm – Deadlock Detection – Deadlock Recovery

Unit IV - Memory Management

Storage Hierarchy – Storage Management Strategies – Contiguous Allocation – Non-Contiguous Allocation – Single User Allocation – Fixed Partition – Variable Partition – Swapping – Virtual Memory (Basic Concepts) – Multilevel Organization – Block Mapping – Paging – Segmentation – Page Replacement Methods – Locality of Reference – Working Set Model

Unit V - I/O and File Systems

I/O Hardware – Device Drivers – Interrupts – DMA – I/O Software – Buffering – Caching – Spooling – Disk Structure – Disk Scheduling Algorithms – File Concept – Access Methods – Directory Structures – File Allocation Methods – Free Space Management – Disk Management – Protection and Security

Text Book:

1. Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, *Operating System Concepts*, 10th Edition, John Wiley & Sons, Inc., 2018.
2. William Stallings, *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson Education Limited, 2018.

Reference Book:

1. William Stallings, *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson Education Limited, 2018.
2. Thomas Anderson and Michael Dahlin, *Operating Systems: Principles and Practice*, 2nd Edition, Recursive Books, 2014.

E Resources:

1. <https://www.perlego.com/book/3866364/silberschatzs-operating-system-concepts-pdf>
2. <https://senabooks.com/ebook/modern-operating-systems-ebook/>

Core Course – VIII: INTERNET AND WEB PROGRAMMING

Syllabus

Unit I: Internet Basics

Introduction to Internet – History of the Internet – Internet Architecture – Client–Server Model – Web Browsers and Web Servers – Domain Name System (DNS) – IP Addressing (IPv4 & IPv6) – Uniform Resource Locator (URL) – Internet Protocols: HTTP, HTTPS, FTP, SMTP

Unit II: HTML and Web Page Design

Introduction to HTML – Structure of HTML Document – Text Formatting Tags – Hyperlinks and Images – Lists and Tables – Forms and Input Controls – Frames and Multimedia Elements – Introduction to HTML5 Features

Unit III: CSS and JavaScript

Introduction to CSS – Types of CSS: Inline, Internal, External – Selectors and Properties – Box Model – Layout Techniques – Introduction to JavaScript – Variables and Data Types – Operators and Control Statements – Functions and Events – Form Validation – DOM Basics

Unit IV: Server-Side Programming

Introduction to Server-Side Scripting – Client-Side vs Server-Side Programming – Introduction to PHP – PHP Syntax and Variables – Control Structures – Functions in PHP – Form Handling using PHP – Sessions and Cookies – File Handling

Unit V: Database and Web Applications

Introduction to Databases – Relational Database Concepts – SQL Basics: CREATE, INSERT, SELECT, UPDATE, DELETE – Connecting PHP with MySQL – CRUD Operations – Introduction to Web Security – Authentication and Authorization – Introduction to Web Hosting and Deployment

Textbooks

1. Jurgen Wolf, HTML and CSS: Visual QuickStart Guide / HTML and CSS, Latest Edition, Rhein Werk Publishing, 2025.
2. Joe Attardi, Modern CSS: Master the Key Concepts of CSS for Modern Web Development, 2nd Edition, A press, 2025.

Reference Books

1. Marijn Haverbeke, Eloquent JavaScript, 4th Edition, No Starch Press, 2024.
2. Julie C. Meloni, HTML, CSS, and JavaScript All-in-One, 4th Edition, Pearson / Sams Teach Yourself, 2022.

E Resources

- i. <https://www.edx.org/learn/web-development>
- ii. <https://nptel.ac.in/courses/106105084>
- iii. <https://www.freecodecamp.org/>

Core Course – IX: SOFTWARE ENGINEERING
Course Code: 24UCS505

Syllabus

Unit I: Overview and Software Process Models

Introduction to Software Engineering – Software Development Process – Software Development Life Cycle (SDLC) – Agile Development Methods – Role of Software Engineers – Software Engineering Practices

Unit II: Requirements and System Design

Requirements Engineering – Software Specification – System Modeling – UML Diagrams – Software Architecture – Design Principles – Coupling and Cohesion – Data Flow Diagrams (DFD) – Object-Oriented Design Concepts

Unit III: Implementation and Testing

Coding Standards and Programming Practices – Software Metrics – Verification and Validation – Unit Testing – Integration Testing – System Testing – Acceptance Testing – Black Box Testing – White Box Testing – Debugging Process – Test Case Design Techniques – Regression Testing – Test Automation

Unit IV: Software Project Management and Quality Assurance

Software Project Planning – Effort Estimation Techniques – Project Scheduling – Risk Management – Project Monitoring and Control – Configuration Management – Software Maintenance – Software Quality Concepts – Quality Assurance Activities – ISO 9001 – CMMI Standards – Review and Audit Process.

Unit V: Modern Software Engineering Practices

Agile Software Development – DevOps Practices – Cloud-based Software Systems – Artificial Intelligence in Software Engineering – Software Engineering for IoT – Software Engineering for Mobile Applications – Ethical and Social Implications in Software Design

Textbook

1. Software Engineering: A Practitioner's Approach – Roger S. Pressman & Bruce R. Maxim – 9th Edition – McGraw-Hill – 2023

Reference Book

1. Software Engineering – Ian Sommerville – 10th Edition – Pearson India – 2018

E – Resources

1. <https://www.ibm.com/topics/software-engineering>
2. <https://www.freecodecamp.org/news/tag/software-engineering/>
3. <https://www.scaler.com/topics/software-engineering/>

Core Course – X: MACHINE LEARNING TECHNIQUES

Course Code: 24UCS506

Credits: 4

Internal marks: 25

Contact Hours/Week: 4

External marks: 75

Syllabus

UNIT – I: INTRODUCTION

Introduction to Machine Learning – Definition and Scope – Types of Machine Learning (Supervised, Unsupervised, Reinforcement Learning) – Applications of Machine Learning – Data Representation and Feature Engineering – Training and Testing Data – Overfitting and Underfitting – Model Evaluation Basics.

UNIT – II: DECISION TREE LEARNING

Decision Tree Representation – Classification and Regression Trees – Entropy and Information Gain – Gini Index – Tree Construction Algorithms – Overfitting in Decision Trees – Pruning Techniques – Advantages and Limitations of Decision Trees.

UNIT – III: PROBABILITY AND BAYES LEARNING

Probability Basics – Conditional Probability – Bayes Theorem – Naïve Bayes Classifier – Bayesian Learning – Probability Distributions – Maximum Likelihood Estimation – Applications of Bayesian Methods in Machine Learning.

UNIT – IV: ARTIFICIAL NEURAL NETWORKS

Biological Neuron vs Artificial Neuron – Perceptron Model – Multilayer Neural Networks – Activation Functions – Forward Propagation – Backpropagation Algorithm – Training Neural Networks – Applications of Neural Networks.

UNIT – V: ENSEMBLES

Introduction to Ensemble Learning – Bagging – Boosting – Random Forest – AdaBoost – Stacking – Voting Methods – Advantages of Ensemble Methods – Applications in Real-world Problems.

Text book

1. *Machine Learning Techniques* – Dr. Rajeev Kapoor & Dimple – First Edition – AG Publishing House (AGPH Books) – 2022

Reference Book

1. *Machine Learning: An Algorithmic Perspective* – Stephen Marsland – 2nd Edition – CRC Press – 2014

E Resource Links:

1. GeeksforGeeks – <https://www.geeksforgeeks.org/machine-learning/>
 2. Scikit-learn – <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>
- Kaggle – <https://www.kaggle.com/learn/intro-to-machine-learning>

Core Course – V: INTERNET AND WEB PROGRAMMING LAB
Course Code: 24UCS553

Credits: 2
Contact Hours/Week: 4

Internal marks: 40
External marks: 60

Syllabus:

1. Explain the components of a URL with an example.
2. Find your system IP address and differentiate IPv4 and IPv6 (any four points).
3. Identify HTTP/HTTPS websites and give one use of FTP and SMTP.
4. Create a basic HTML page with title, heading, and paragraph.
5. Demonstrate HTML text formatting tags.
6. Create a webpage with a hyperlink and a clickable image.
7. Develop an HTML form with Name, Email, and Password fields.
8. Demonstrate Inline, Internal, and External CSS.
9. Design a webpage to illustrate the CSS box model.
10. Write a JavaScript program to display variables using alert.
11. Implement JavaScript form validation for Name and Email.
12. Write a PHP program for Hello World and addition of two numbers.
13. Create a form and display submitted data using PHP.
14. Write SQL queries for creating, inserting, and selecting student records.
15. Write a PHP program to connect MySQL and perform insert and display operations.

Discipline Centric Elective Course I: SOFT COMPUTING TECHNIQUES

Course Code: 24UCS701

Credits: 4

Contact Hours/Week: 4

Internal marks: 25

External marks: 75

Syllabus:

UNIT – I: Fundamentals of Soft Computing

Introduction to Soft Computing – Concept of Soft Computing – Characteristics of Soft Computing – Hard Computing vs Soft Computing – Components of Soft Computing: Neural Networks, Fuzzy Logic, Evolutionary Algorithms – Approximate Reasoning and Uncertainty Handling – Applications of Soft Computing

UNIT – II: Artificial Neural Networks

Introduction to Artificial Neural Networks – Biological Neuron vs Artificial Neuron – Structure of Neural Networks – Activation Functions – Single Layer Networks – Multi-Layer Networks – Learning Methods: Supervised Learning and Unsupervised Learning – Backpropagation Algorithm – Applications of Artificial Neural Networks

UNIT – III: Fuzzy Logic Systems

Introduction to Fuzzy Logic – Crisp Sets vs Fuzzy Sets – Membership Functions – Linguistic Variables – Fuzzy Set Operations – Fuzzy Rules and Rule Base – Fuzzy Inference Systems: Mamdani Model and Sugeno Model – Defuzzification Methods – Applications of Fuzzy Logic

UNIT – IV: Evolutionary Computing (Genetic Algorithms)

Introduction to Genetic Algorithms – Basic Concepts: Chromosomes, Genes, Population – Fitness Function – Selection Operator – Crossover Operator – Mutation Operator – Genetic Algorithm Procedure – Optimization using Genetic Algorithms – Applications of Genetic Algorithms

UNIT – V: Hybrid Soft Computing Systems and Applications

Introduction to Hybrid Soft Computing Systems – Neuro-Fuzzy Systems – Genetic Neural Networks – Integration of Soft Computing Techniques – Applications in Data Mining – Applications in Image Processing – Applications in Decision Systems – Advantages of Soft Computing – Limitations of Soft Computing – Case Studies

Textbook

1. *Soft Computing: Integrating Evolutionary, Neural and Fuzzy Systems*, Andrea Tettamanzi and Marco Tomassini published by Springer (Springer-Verlag Berlin Heidelberg) 2001.

Reference Book

1. S. Rajasekaran and G. A. Vijayalakshmi Pai authored the book *Neural Networks, Fuzzy Systems and Evolutionary Algorithms: Synthesis and Applications*, published by PHI 2003

E-Resources

1. https://books.google.com/books/about/Soft_Computing.html?id=86jQw6v30KoC

Elective Laboratory I: SOFT COMPUTING TECHNIQUES LAB
Course Code: 24UCS751

Credits: 2
Contact Hours/Week:4

Internal marks: 40
External marks: 60

Syllabus

1. Install Python and required libraries (NumPy, Matplotlib) and write a program to perform basic numerical operations and plot a simple graph.
2. Write a Python program to demonstrate basic matrix operations (addition, multiplication, transpose) using NumPy.
3. Write a program to visualize simple data distributions using graphs (line, bar, scatter) with Matplotlib.
4. Write a program to implement a simple artificial neuron using weighted sum and activation function.
5. Write a program to plot different activation functions such as Sigmoid, ReLU, and Tanh.
6. Develop a program to implement a single-layer perceptron to solve AND/OR logic gate problems.
7. Implement a multi-layer neural network using any library (TensorFlow/Keras) to perform simple classification.
8. Write a program to train a neural network using the backpropagation algorithm for a given dataset.
9. Create and plot different types of membership functions such as triangular, trapezoidal, and Gaussian.
10. Write a program to perform fuzzy set operations like union, intersection, and complement.
11. Design and implement a simple fuzzy inference system for a real-world problem (e.g., temperature control).
12. Implement a simple genetic algorithm to find the maximum value of a given function.
13. Write a program to demonstrate genetic algorithm operators such as selection, crossover, and mutation.
14. Develop a program to solve an optimization problem (e.g., shortest path or function optimization) using a genetic algorithm.

Discipline Centric Elective Course I: Introduction to GITHUB

Course Code: 24UCS703

Credits: 4

Contact Hours/Week: 4

Internal marks: 25

External marks: 75

Syllabus:

Unit I - Introduction

Centralized Version Control: Version Control System – Basics with Supervision – Need and Importance in Software Development – Problems without Version Control – Versioning Concept – Repository – Working Directory and Staging Area – History of Version Control System

Unit II - Distributed Control System

Types of Version Control Systems – Local Version Control System – Working of Centralized Version Control System – Distributed Version Control System – Working Comparison of CVCS and DVCS – Architecture of VCS – Client–Server Model and Distributed Model – Data Storage, Access and Sharing Mechanisms – Advantages and Limitations of VCS

Unit III - Git and Basic Operations

Introduction to Git – Features and Advantages of Git – Installation and Initial Setup – Repository Creation and Initialization – Basic Commands: add, commit, status, log – File Tracking and Version History – Remote Commands: push, pull, clone – Git Workflow and Lifecycle – Understanding Commit History and Logs

Unit IV - Branching and Collaboration

Concepts and Types of Branches – Creating, Switching and Managing Branches – Merging Techniques and Strategies – Conflict Identification and Resolution – Version History and Log Management – Collaborative Development using Version Control Systems – Managing Multiple Contributors and Team Workflows

Unit V - Remote Repositories and Advanced Concepts

Introduction to Remote Repository Concepts and Hosting Platforms – Working with Remote Repositories – Push, Pull and Fetch Operations – Pull Requests and Code Review Process – Issue Tracking and Project Management Tools – Version Control Best Practices – Backup, Recovery and Security in VCS – Real-world Applications and Case Studies

Textbook:

1. “Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development” – Jon Loeliger & Matthew McCullough, O’ Reilly Media

Reference Book:

1. Jon Loeliger and Matthew McCullough, *Version Control with Git*, 2nd Edition, O’Reilly Media, 2012.
2. Scott Chacon and Ben Straub, *Pro Git*, 2nd Edition, Apress, 2014.
3. Prem Kumar Ponuthorai and Jon Loeliger, *Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development*, 2nd Edition, O’Reilly Media, 2012

E Resources:

<https://www.tutorialspoint.com/git/index.htm> <https://www.coursera.org/learn/version-control-with-git>
<https://www.geeksforgeeks.org/git/git-tutorial/>

Discipline Centric Elective Laboratory I : GITHUB LAB

Course Code: 24UCS752

Credits: 2

Contact Hours/Week:4

Internal marks: 40

External marks: 60

Syllabus

1. Implement Git installation and set up user credentials in the system.
2. Execute repository initialization and handle file tracking using git add, git commit, and git status.
3. Examine and utilize version tracking through git log and git diff to monitor changes.
4. Demonstrate branching operations by creating, switching, and organizing branches.
5. Integrate changes by performing branch merging techniques.
6. Resolve merge conflicts using appropriate conflict management strategies.
7. Set up and establish a GitHub account and create a remote repository.
8. Configure remote repository connections using git remote add.
9. Perform synchronization of repositories using push, pull, and fetch operations.
10. Replicate and modify repositories by cloning and editing project files.
11. Develop a collaborative workflow by implementing forking and pull request mechanisms.
12. Construct project documentation using README and .gitignore files.
13. Manage version releases by applying tagging techniques.
14. Design a structured branching strategy for efficient team-based development.
15. Build a complete mini project utilizing GitHub collaboration and version control practices.

Discipline Centric Elective Course I
USER INTERFACE AND USER EXPERIENCE DESIGN
Course Code: 24UCS705

Credits:4
Contact Hours/Week: 4

Internal marks: 25
External marks: 75

Syllabus

Unit I – User Interface Introduction

Definition of User Interface – Definition of User Experience – Importance of UI/UX – Difference between UI and UX – History and Evolution of UI/UX – Characteristics of Good Design – Principles of UI Design – Consistency, Feedback, Visibility, Simplicity – Human-Centered Design – Usability Concepts – Accessibility Basics

Unit II - User Research and Analysis

User Research – Types of Users – Data Collection Methods – Surveys and Interviews – Observation Techniques – Personas – Empathy Mapping – User Journey Mapping – Task Analysis – Scenario Building – Requirements Gathering – User Needs Analysis

Unit III - Interaction Design

Interaction Design Principles – Human-Computer Interaction (HCI) Basics – Information Architecture – Navigation Design – Wireframing – Prototyping – Low-Fidelity and High-Fidelity Designs – Usability Principles – Learnability, Efficiency, Memorability – Error Handling and Feedback

Unit IV - Visual Design and UI Elements

Visual Design Elements – Color Theory – Typography – Layout and Grid Systems – Icons and Images – Design Systems – Style Guides – Responsive Design – Accessibility Guidelines – UI Components (Buttons, Forms, Menus, Dialogs) – UI Design Tools (Figma, Adobe XD, Sketch)

Unit V - Testing, Evaluation and Future Trends

Usability Testing – Types of Usability Testing – Moderated and Unmoderated Testing – Remote and In-Person Testing – A/B Testing – Heuristic Evaluation – Nielsen's Usability Heuristics – User Testing Methods – Think-Aloud Protocol – Eye Tracking and Heatmaps – Feedback Collection Methods

Textbook:

1. Jeff Johnson, *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*, 3rd Edition, Morgan Kaufmann (Elsevier), 2020.
2. “The UX Book: Agile UX Design for a Quality User Experience” – Rex Hartson & Pardha Pyla, Elsevier

E Resources:

<https://www.mygreatlearning.com/academy/learn-for-free/courses/ui-ux> <https://alison.com/course/introduction-to-ui-ux-design>

Discipline Centric Elective Course
User Interface and User Experience Design Lab
Course Code: 24UCS753

Credits: 2

Contact Hours/Week: 4

Internal marks:40

External marks: 60

Syllabus

1. Examine and compare the UI/UX design of two mobile applications and recommend improvements.
2. Conduct a user survey and interpret the results to identify user requirements.
3. Prepare an empathy map for a selected group of users.
4. Break down and analyze the steps involved in completing a task in an application.
5. Improve the navigation structure of an existing website using design principles.
6. Redesign a given interface by using appropriate color and typography principles.
7. Perform usability testing on an application and interpret the findings.
8. Demonstrate the use of UI design principles such as consistency and feedback in a sample interface.
9. Identify usability issues in a website and suggest suitable solutions.
10. Modify an interface by incorporating accessibility guidelines.
11. Analyze user behavior based on observation and present your findings.
12. Organize website content effectively using information architecture principles.
13. Evaluate an application based on usability factors like learnability and efficiency.
14. Improve error handling and feedback mechanisms in an existing UI design.
15. Examine a responsive website and assess its performance across different devices.

Discipline Centric Elective Course I
NOSQL AND MONGODB
Course Code: 24UCS707

Credits: 4

Contact Hours/Week: 4

Internal marks: 25

External marks: 75

Syllabus:

UNIT - I: Fundamentals of NoSQL Databases

Introduction – Traditional RDBMS Limitations – What is NoSQL – Characteristics of NoSQL Databases – Types of NoSQL Databases: Key-Value, Document, Column-Family, Graph – CAP Theorem – BASE Properties vs ACID Properties – Advantages and Disadvantages of NoSQL Databases

UNIT - II: MongoDB Architecture and Data Modeling

MongoDB Introduction – Features of MongoDB – JSON and BSON Data Formats – Documents and Collections – MongoDB Architecture – Installation and Setup – MongoDB Shell and Compass – Data Modeling Concepts – Embedding vs Referencing

UNIT - III: MongoDB CRUD Operations and Querying

CRUD Operations – Insert Documents – Query Documents using find() – Update Operations – Delete Operations – Query Operators: Comparison, Logical, Element – Projection – Sorting and Limiting Results – Working with Arrays and Embedded Documents

UNIT - IV: Indexing, Aggregation and Performance Optimization

Indexing – Types of Indexes: Single Field, Compound, Multikey – Index Management – Aggregation Framework – Aggregation Pipeline Stages – MapReduce Concept – Performance Optimization – Query Optimization Techniques.

UNIT - V: MongoDB Scalability, Security and Applications

Replication – Replica Sets – Sharding – Data Distribution in MongoDB – Load Balancing – Security Concepts – Authentication and Authorization – MongoDB Atlas – Backup and Recovery – Real-time Applications of MongoDB

Text Book:

1. MongoDB: The Definitive Guide — Authors: Kristina Chodorow and Michael Dirol Publisher: O'Reilly Media, Inc. Year: 2010.

Reference Books:

1. MongoDB 8.0 in Action (3rd Edition) — Author: Arek Borucki. Publisher: Manning Publications. Year: 2025.
2. Mastering MongoDB 7.0 (4th Edition) — Authors: Marko Aleksendrić, Arek Borucki & Leandro Domingues. Publisher: Packt Publishing. Year: 2024.
3. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence — Authors: Martin Fowler & Pramod J. Sadalage. Publisher: Addison-Wesley Professional. Year: 2012.
4. Designing Data-Intensive Applications — Author: Martin Kleppmann.

Publisher: O'Reilly Media, Inc. Year: 2017.

E- Resources:

1. Official Documentation: MongoDB Inc. — <https://www.mongodb.com/docs/>
2. Online Learning Platform: MongoDB University — <https://university.mongodb.com/>
3. Tutorials and Guides: GeeksforGeeks — <https://www.geeksforgeeks.org/mongodb/>

Discipline Centric Elective Course I
NOSQL AND MONGODB LAB
Course Code: 24UCS754

Credits: 2
Contact Hours/Week: 4

Internal marks: 40
External marks: 60

Syllabus

1. Illustration of Where Clause, AND,OR operations in MongoDB.
2. Execute the Commands of MongoDB and operations in MongoDB : Insert, Query, Update, Delete and Projection. (Note: use any collection).
3. Develop a MongoDB query to select certain fields and ignore some fields of the documents from any collection.
4. Develop a MongoDB query to display the first 5 documents from the results obtained in a. [use of limit and find].
5. Execute query selectors (comparison selectors, logical selectors) and list out the results on any collection
6. Execute query selectors (Geospatial selectors, Bitwise selectors) and list out the results on any collection.
7. Create and demonstrate how projection operators (\$, \$elematch and \$slice) would be used in the MondoDB.
8. Execute Aggregation operations (\$avg, \$min,\$max, \$push, \$addToSet etc.). students encourage to execute several queries to demonstrate various aggregation operators).
9. Execute Aggregation Pipeline and its operations (pipeline must contain \$match, \$group, \$sort, \$project, \$skip etc. students encourage to execute several queries to demonstrate various aggregation operators)
10. Find all listings with listing_url, name, address, host_picture_url in the listings And Reviews collection that have a host with a picture url
11. Using E-commerce collection write a query to display reviews summary.
12. Demonstrate creation of different types of indexes on collection (unique, sparse, compound and multikey indexes) using indexes. b. Demonstrate optimization of queries
13. Develop a query to demonstrate Text search using catalog data collection for a given word.
14. Develop queries to illustrate excluding documents with certain words and phrases
15. Develop an aggregation pipeline to illustrate Text search on Catalog data collection.

24UGP791: Educational Psychology

Credits: 4

Contact Hours: 4

Internal Marks: 25

External Marks: 75

Unit I - Perception and perceptual processes: Meaning and nature of perception- approaches to visual perception- bottom-up approach- top-down approach- theories of visual perception – Gibson’s theory of direct perception- the template matching theory- feature-matching theory- Gregory’s theory of perception and approach & principles of perceptual organisation- perception of distance, direction, constancy& illusions

Unit II - Attention: Meaning and nature- signs and effects of attention- factor or determinants of attention- classification of attention- selective attention-meaning and nature- theories of selective attention- automatic and controlled process in attention-controlled processing in attention – executive attention- influencing sustained attention-theories of sustained attention- divided attention- the phenomenon of shifting –span of attention

Unit III – Learning meaning Process and Theories: meaning and definition of the term learning- the process of learning- theories of learning- Thorndike’s connectionism- laws of learning- Watson and Pavlov’s classical conditioning-skinner operant conditioning- Guthrie’s theory of Continuous conditioning- Hull’s systematic Behaviour theory- Gestalt theory of Insightful learning – Toman’s sign learning

Unit IV - Memory: Meaning and nature- process of memorisation- encoding storage and retrieval- the of memory- models of memory-Atkinson and Shiffrins model- Miller’s information processing – Dual coding model- Baddeley’s model- Tulving multiple memory systems model- Parallel distribution processing models-long term memory-measurement of memory

Unit V - Language and language process: Introduction- Human Language – characteristic Features – forms of Language-origin of language- Neurological Basis of language- language structure- comprehension- thought- development of Oral vocabulary- Oral expression or speech power- reading ability- theories of language development

24UCP758: Teaching Computer Science

Credits: 2

Contact Hours: 4

Internal Marks: 40

External Marks: 60

Syllabus:

Module 1: Foundations of Teaching & Learning (Weeks 1–2) Topics:

- Role of a Computer Science teacher
- Characteristics of effective teaching
- Understanding learners (school-level psychology basics)
- Learning theories (behaviorism, constructivism, experiential learning)

Practical Activities:

- Classroom observation (video/live)
- Reflection writing on teaching styles

Outcomes:

Students understand the basics of teaching and learner behavior.

Module 2: Computer Science Pedagogy (Weeks 3–4) Topics:

- Pedagogical approaches for teaching CS:
 - Demonstration method
 - Activity-based learning
 - Problem-solving method
 - Project-based learning
- Teaching programming concepts to beginners
- Addressing common student misconceptions

Practical Activities:

- Micro-teaching sessions (5–10 minutes per student)
- Peer feedback sessions

Outcomes:

Students learn how to effectively teach CS concepts using varied methods.

Module 3: Lesson Planning & Instructional Design (Weeks 5–6) Topics:

- Components of a lesson plan
- Writing learning objectives (Bloom's Taxonomy)
- Structuring:
 - Introduction
 - Presentation
 - Practice
 - Assessment

- Teaching plan vs lesson plan

Practical Activities:

- Prepare lesson plans for:
 - Basic programming
 - Computer fundamentals
- Group review and improvement

Outcomes:

Students can design structured and outcome-based lesson plans.

Module 4: Teaching Aids & Technology Integration

(Weeks 7–8) Topics:

- Use of digital tools in teaching:
 - Presentation tools (PowerPoint, Canva)
 - Coding platforms (Scratch, online compilers)
- Smart classroom tools
- Multimedia usage (videos, animations, simulations)
- Gamification in learning

Practical Activities:

- Create teaching content (PPT/video/demo)
- Design interactive classroom activities

Outcomes:

Students can enhance teaching using modern technological tools.

Module 5: Classroom Management & Engagement (Weeks

9–10) Topics:

- Classroom management strategies
- Handling diverse learners
- Encouraging student participation
- Questioning techniques
- Managing practical lab sessions

Practical Activities:

- Role-play classroom scenarios
- Conduct mock classroom sessions

Outcomes:

Students gain confidence in handling real classroom environments.

Module 6: Assessment & Evaluation Techniques (Weeks 11–12) Topics:

- Types of assessment:
 - Formative vs summative
- Designing quizzes, assignments, and practical tests
- Rubrics for evaluation
- Providing constructive feedback

Practical Activities:

- Create question papers and rubrics

- Evaluate peer teaching sessions

Outcomes:

Students learn how to assess learning effectively.

Module 7: Learning Management Systems (LMS) (Weeks 13–14) Topics:

- Introduction to LMS (e.g., Moodle, Google Classroom)
- Creating courses and uploading content
- Managing assignments and grading
- Online communication and discussion forums

Practical Activities:

- Create a course in LMS
- Upload lesson materials, quizzes, assignments

Outcomes:

Students can use LMS platforms to manage digital learning.

Module 8: Teaching Practice & Capstone (Weeks 15) Topics:

- Full-length teaching practice
- Integration of pedagogy + technology + planning

Practical Activities:

- Conduct complete teaching sessions (20–30 mins)
- Submit:
 - Lesson plan
 - Teaching aids
 - Assessment tools
- Internship / school teaching exposure (if possible)

Outcomes:

Students demonstrate complete teaching competency.

Ability Enhancement - III : Multimedia and Animation Lab
Course Code: 24UCS381

Syllabus:

1. Create formatted text using different fonts and styles.
2. Design a poster using shapes, colors, and typography.
3. Perform basic image editing (crop, resize, brightness, contrast).
4. Create a logo using vector graphics tools.
5. Apply filters and effects to enhance images.
6. Design a banner using layers and blending techniques.
7. Create a photo collage using masking techniques.
8. Record and edit audio (cut, trim, apply effects).
9. Create a video using images, text, and audio.
10. Add transitions, subtitles, and background music to a video.
11. Prepare a storyboard for animation.
12. Create a 2D animation using basic techniques.
13. Design and animate a simple character.
14. Create motion graphics using text and effects.
15. Develop a mini multimedia project integrating all components