Proposed Syllabus (Semester-I to Semester-VIII)

For

B.Sc. in Computer Science
(Major / Honours / Honours with Research)

Submitted

To

University of Gour Banga

Under
New Education Policy-2020
[With effect from the Session 2024 – 25 **]**

Contents

Abbreviations:

DC-MJ: Discipline Specific Course - Major Core
DC-MN: Discipline Specific Course - Minor Core
MDC: Multidisciplinary Course.
AEC: Ability Enhancement Course.
SEC: Skill Enhancement Course.
IAPWC: Internship / Apprenticeship / Project / Workshop / Community Outreach Course.
VAC: Value-Added Course.
FM: Full Marks

Note:
*Students with Computer Science as **major** will follow the **DSC** paper for the concerned semester from this syllabus. Students with major have to choose one subject as minor from the respective pool as per University guidelines. MDC, AEC, SEC, IAPWC and VAC papers are compulsory, major students has to choose it as per the available guidelines provided by the Institution / University.*

Credit Distribution

Sem	Major Core (MJ)	Multi-disciplinary (MDC)	Minor Core (MN)	Ability Enhance Course (AEC)	SEC	IAPWC	VAC	Research Project / Dissertation	Total Credit
I	**DC-MJ-101** (4)	**MDC-101** (3)	**DC-MN-101** (4)	**MIL-1** (2)	**SEC-101** (3)	**IAPWC-101** (2)	**VAC-101** (2) ENVS	-	**20**
II	**DC-MJ-201** (4)	**MDC-201** (3)	**DC-MN-201** (4)	**MIL-2** (2)	**SEC-201** (3)	**IAPWC -201** (2)	**VAC-102** (2) Choose from pool	-	**20**
III	**DC-MJ-301** (4) **DC-MJ-302** (4)	**MDC-301** (3)	**DC-MN-301** (4)	**English language-1** (2)	**SEC-301** (3)	**IAPWC -301** (2)		-	**22**
IV	**DC-MJ-401** (4) **DC-MJ-402** (4) **DC-MJ-403** (4)	-	**DC-MN-401** (4)	**English Language- 2** (2)	-		**VAC-103** (2) Choose from pool	-	**20**
V	**DC-MJ-501** (4) **DC-MJ-502** (4) **DC-MJ-503** (4) **DC-MJ-504** (4)	-	**DC-MN-501** (4)	-	-		-	-	**20**
VI	**DC-MJ-601** (4) **DC-MJ-602** (4) **DC-MJ-603** (4) **DC-MJ-604** (4)	-	**DC-MN-601** (4)	-	-		-	-	**20**
For Honours									
VII	**DC-MJ-701** (4) **DC-MJ-702** (4) **DC-MJ-703** (4) **DC-MJ-704** (4)	-	**DC-MN-701** (4)	-	-	-	-	-	**20**
VIII	**DC-MJ-801** (4) **DC-MJ-802** (4) **DC-MJ-803** (4) **DC-MJ-804** (4)	-	**DC-MN-801** (4)	-	-	-	-	-	**20**
For Honours with Research									
VII	**DC-MJ-701** (4) **DC-MJ-702** (4) **DC-MJ-703** (4) **DC-MJ-704** (4)	-	**DC-MN-701** (4)	-	-	-	-	-	**20**
VIII	**DC-MJ-801** (4) **DRP-1** (12)	-	**DC-MN-801** (4)	-	-	-	-	-	**20**

[**Note:** Total Credit Point has been mentioned in the bracket.]

Computer Science Course Structure

Sem	Course Code	Course Title	Lab	Credit	Marks Distribution
I	DC-MJ-101	Introduction to Programming using C	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
II	DC-MJ-201	Digital Logic System	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
III	DC-MJ-301	Object Oriented Programming through C++	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-302	Computer System Organization and Architecture	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
IV	DC-MJ-401	Discrete Mathematics	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-402	Data Structures and Algorithms using C++	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-403	Operating System	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
V	DC-MJ-501	Theory of Computation and Compiler Design	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-502	Computer Graphics using OpenGL	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-503	Advanced Database Management System	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-504	Programming through Python	PR	4	50 (PR) + 25 (IA) = **75**
VI	DC-MJ-601	Machine Learning	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-602	Data Analytics using Python	PR	4	50 (PR) + 25 (IA) = **75**
	DC-MJ-603	Data Communication and Networking	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-604	Minor Project	PR	4	50 (PR) + 25 (IA) = **75**
For Honours					
VII	DC-MJ-701	Advanced Design and Analysis of Algorithm	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-702	Software Engineering	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-703	Artificial Intelligence	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-704	Research Methodology	PR	4	50 (PR) + 25 (IA) = **75**
VIII	DC-MJ-801	Deep Learning	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-802	Computational Intelligence	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-803	Web Programming	PR	4	50 (PR) + 25 (IA) = **75**
	DC-MJ-804	Major Project	PR	3+1=4	50 (Preparation) + 25 (Presentation / Viva-Voce) = **75**
For Honours with Research					
VII	DC-MJ-701	Advanced Design and Analysis of Algorithm	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DC-MJ-702	Software Engineering	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-703	Artificial Intelligence	TH	4	50 (TH) + 25 (IA) = **75**
	DC-MJ-704	Research Methodology and Application	PR	4	50 (PR) + 25 (IA) = **75**
VIII	DC-MJ-801	Deep Learning	TH+PR	3+1=4	30 (TH) + 20 (PR) + 25 (IA) = **75**
	DRP-1	Dissertation	PR	8+4=12	150 (Preparation) + 75 (Presentation / Viva-Voce)= **225**

*TH, PR, and IA stands for Theory, Practical, and Internal Assessment respectively.

Descriptive Type Question Patterns for Discipline Specific Course - Major Core (DC-MJ)

- **Theory (Semester End Written Examination) -TH**

  Group-A **Group-B**

  Full Marks = 30 (2 Marks x 5 Question) + (5 Marks x 4 Questions)
  Full Marks = 50 (2 Marks x 5 Question) + (10 Marks x 4 Questions)

  Note: Question(s) containing 10 marks will be divided into smaller sub-parts with maximum mark - 5.
  At-least one extra question will be available for each group to answer.

- **Practical (Semester End Laboratory Based Test) - PR**

  Full Marks = 20 / 25/ 50 / 75
  [Distribution of practical examination marks have been mentioned in the concerned part of the Syllabus.]

- **Internal Assessment - IA**

  Full Marks = 25 [Attendance (5) + Classroom Performance (5) + Presentation/project etc. (10) + Viva-voce (5)]

- **Duration of Examination**

  - Theory paper of 30 / 40 / 50 marks: 2 hours
  - Practical paper of 20 / 25 marks: 2 hours
  - Practical paper of 50 / 75 marks: 3 hours

Semester-I

DC-MJ-101(TH): Introduction to Programming using C

Introduction: Basic Structure, Algorithms, Flowcharts, Structured programming constructs.
C Programming elements: Character sets, Keywords, Constants, Variables, Data Types, Operators- Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional Operator, Precedence and Associations; Expressions, type casting. Comments, Functions, Storage Classes, Bit manipulation, Input and output, Command Line Arguments.
C Pre-processor: File inclusion, Macro substitution. **Statements:** Assignment, Control statements- if, if else, switch, break, continue, goto, Loops-while, do_while, for. **Functions:** Argument passing, return statement, return values and their types, recursion **Arrays:** String handling with arrays, String handling functions.1D Arrays, 2D Arrays with static and dynamic memory allocations. **Pointers:** Definition and initialization, Pointer arithmetic, Pointers and arrays, Function pointers, String functions and manipulation, Dynamic storage allocation and the types, Dangling pointers. **User defined Data types:** Structures. Structure arrays, Pointers to Functions and Structures, Unions **File Access:** Opening, Closing, I/O operations.

DC-MJ-101(PR): Introduction to Programming using C Lab

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C compiler **(GCC)** for practical.

1. Write a program to print the sum and product of digits of an integer.
2. Write a program to reverse a number.
3. Write a program to compute the sum of the first 'n' terms of the following series, S =1+1/2+1/3+1/4+……
4. Write a program to compute the sum of the first 'n' terms of the following series, S =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. Write a program to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. Write a program to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
```

10. Write a program to perform following actions on an array entered by the user : i) Print the even-valued elements. ii) Print the odd-valued elements. iii) Calculate and print the sum and average of the elements of array.
iv) Print the maximum and minimum element of array v) Remove the duplicates from the array vi) Print the array in reverse order The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
16. Write a menu driven program to perform following operations on strings: a) Show address of each character in string. b) Concatenate two strings without using strcat function. c) Concatenate two strings using strcat function. d) Compare two strings. e) Calculate length of the string (use pointers). f) Convert all lowercase characters to uppercase. g) Convert all uppercase characters to lowercase. h) Calculate number of vowels. i) Reverse the string.
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. Write a program to display Fibonacci series (i)using recursion, (ii) using iteration.
19. Write a program to calculate Factorial of a number (i)using recursion, (ii) using iteration.
20. Write a program to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

Text/ Reference Books:

1. Programming with C, Byron S. Gottfried, McGraw Hill.
2. The C Programming Language, Kernighan and Dennis, PHI.
3. The Complete reference C, Herbert Schildt, McGraw Hill.
4. Let Us C, Kanitkar, BPB Publication.
5. Programming in ANSI C, Balaguruswamy, McGraw Hill.
6. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

Note: *Student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 6*
c. *Output – 4*
d. *Viva- Voce – 5*
e. *Lab Assignment – 3*

Semester – II

DC-MJ-201(TH): Digital Logic System

Boolean Algebra: Fundamentals of Boolean Expression: Definition of Switching Algebra, Basic properties of Switching Algebra, Huntington's Postulates, Basic logic gates (AND, OR, NOT), De-Morgan's Theorem, Universal Logic gates (NAND, NOR), Minterm, Maxterm, Minimization of Boolean Functions using K-Map, Simplification of logic expression.
Combinational Circuits: Half adders, Full Adder, Half Subtractor, Full Subtractor and construction using Basic Logic Gates (OR, AND, NOT) and Universal Logic Gates (NAND & NOR), Multibit Adder- Ripple Carry Adder, Carry Look Ahead adder, BCD Adder, Adder/Subtractor unit Construction using 4 bit Full adders units, 1 bit, 2 bit and 3 bit Comparators. Data Selector-Multiplexer: Expansion (Cascading), Function Realization. Encoders:- Realization of simple Encoders and priority Encoders using Basic and Universal Logic gates. Data Distributor:- De-multiplexer, Cascading. Chip Selector/Minterm Generator - Decoder-Function Realization, Cascading, BCD Decoders, Seven Segment Display and Decoders, realization of seven segment decoders using basic gates. Parity bit and Code Converters: Parity bit Generator/Checker, Gray to Binary code converter, Binary to Gray Code Converter.

Sequential Circuits: Set/Reset (SR) Latch: Using NAND and NOR gates, Gated S-R latches, D Latch, J-K Latch, T Flip Flop, Race around Condition, Master Slave J-K Flip Flop, Edge Triggered SR, D and JK Flip Flop, Flip-Flop Conversions, Flip-Flops with Preset and Clear.
Registers: Serial Input Serial Output, Serial Input Parallel Output, Parallel input Serial Output, Parallel Input parallel Output, Universal Shift Registers.
Counters: Asynchronous Counter: UP/DOWN Counters, Mod - N Counters, BCD Counter, Synchronous Counter: UP/DOWN Counters, Mod-N Counters, Ring Counters, Johnson Counters.

DC-MJ-201(PR): Digital Logic System Lab

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

Combinational Circuits:
1. Implementation of different functions using Basic and Logic gates, SOP, POS.
2. Study and prove De-Morgan's Theorem.
3. Universal function using NAND and NOR gates
4. Implementation of half and Full adder (3-bit) using basic logic gates and Universal logic gates (NAND & NOR).
5. Implementation of half and Full Subtractor (3-bit) using basic logic gates and Universal logic gates (NAND & NOR).
6. BCD adder using 7483 and other logic gates.
7. Design 4 to 1 multiplexer using logic/Universal gates and implement full adder/full subtractor.
8. Using 74153 and 74151 to implement full adder/ full subtractor and other functions.
9. Cascading of Multiplexers.
10. Design 2 to 4 decoder using basic / universal logic gates.
11. Study 74138 and 74139 and implement full adder / full subtractor and other functions.
12. Implementation of 1 bit Comparator using decoders.
13. Cascading of Decoders.
14. Design a parity generator and checker using basic gates.
15. Construct and study comparators using 7485.
16. Construct Comparator (2-bit) using logic gates
17. Design a seven segment display unit using Common anode/Common cathode and 7447 / 7448.
18. Study Priority Encoder Chip 74147/74148.

Sequential Circuits:
1. Realization of RS, D, JK Clocked/Gated Level Triggered Flip-Flop using basic/Universal logic gates.
2. Study and Conversion of Flip-Flops: D to JK, JK to D, JK to T, SR to JK, SR to D Flip-flop.
3. Design synchronous and asynchronous counters MOD-n (MOD-8, MOD-10) UP/ DOWN and connecting Seven Segment Display along with decoder for display of counting sequence.
4. Construction of ODD/EVEN 4 bit Synchronous Counter.
5. 2-bit Universal Shift Register.

Text/Reference Books:

1. Digital Circuits, Vol - I & II, D. Ray Chaudhuri, Platinum Publishers.
2. Digital Systems - Principle & Applications, Tocci & Widmer, EEE.
3. Digital Logic & State Machine Design, Comer, Oxford.
4. Digital Principle & Applications, Malvino & Leach, McGraw Hill.

5. Digital Design, Mano, PHI.
6. Digital Integrated Electronics- H. Taub & D. Shilling, Mc Graw Hill.
7. Digital Circuits and Design, Salivahan, Vikas.

Note: *Student must be familiar with Trainer-kit as well as separate breadboard, power supply, LED, Resistance, Clock based individual system. For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Circuit Design – 4*
b. *Implementation – 8*
c. *Viva- Voce – 5*
d. *Lab Assignment – 3*

Semester – III

DC-MJ-301(TH): Object Oriented Programming through C++

Introduction to C++: Overview of Procedural Programming and Object-Orientation Programming, Using main() function, Compiling and Executing Simple Programs in C++. Concepts of Data Types, Variables, Constants, Operators and Basic I/O Expressions, Conditional Statements and Iterative Statements, Functions and Arrays Pointers and References in C++, Memory Allocation in C++.
Using Classes in C++: Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables &Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors.
Overview of Function Overloading and Operator Overloading: Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (unary and binary operators)
Inheritance, Polymorphism, Exception Handling and Generic function: Introduction to Inheritance (Single, Multi-Level, Multiple, Hierarchical, Hybrid), Polymorphism (Virtual Functions, Pure Virtual Functions), Abstract class, Basics of Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions. Generic function-Template class and function.

DC-MJ-301(PR): Object Oriented Programming through C++ Lab

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C++ compiler **(GNU C++)**.

1. Write a program to print the sum and product of digits of an integer.
2. Write a program to reverse a number.
3. Write a program to compute the sum of the first 'n' terms of the following series S = 1+1/2+1/3+1/4+……
4. Write a program to compute the sum of the first 'n' terms of the following series S =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. Write a program to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. Write a program to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
*******
```

10. Write a program to perform following actions on an array entered by the user: i. Print the even-valued elements ii. Print the odd-valued elements iii. Calculate and print the sum and average of the elements of array iv. Print the maximum and minimum element of array v. Remove the duplicates from the array vi. Print the array in reverse order The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using new operator.
16. Write a menu driven program to perform following operations on strings: a) Show address of each character in string b) Concatenate two strings without using strcat function. c) Concatenate two strings using strcat function. d) Compare two strings e) Calculate length of the string (use pointers) f) Convert all lowercase characters to uppercase g) Convert all uppercase characters to lowercase
h) Calculate number of vowels i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. Write a program to display Fibonacci series (i)using recursion, (ii) using iteration
19. Write a program to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. Write a program to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose

22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
24. Create a class Box containing length, breath and height. Include following methods in it: a) Calculate surface Area b) Calculate Volume c) Increment, Overload ++ operator (both prefix & postfix) d) Decrement, Overload -- operator (both prefix & postfix) e) Overload operator == (to check equality of two boxes), as a friend function f) Overload Assignment operator g) Check if it is a Cube or cuboid Write a program which takes input from the user for length, breath and height to test the above class.
25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
26. Write a program to retrieve the student information from file created in previous question and print it in following format: Roll No. Name Marks
27. Copy the contents of one text file to another file, after removing all whitespaces.
28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

Text/ Reference Books:

1. C++: The Complete Reference, Herbtz Schildt, McGraw Hill.
2. The C++ Programming Language, Bjarne Stroustrup, Addison-Wesley.
3. Programming -- Principles and Practice using C++,Bjarne Stroustroup, Addison-Wesley.
4. Object Oriented Programming with C++, E Balaguruswamy, Tata McGraw-Hill Education.
5. C++ How to Program, Paul Deitel, Harvey Deitel, Prentice Hall.
6. Programming with C++, John R. Hubbard, Schaum's Series.
7. Accelerated C++, Andrew Koeni, Barbara, E. Moo, Published by Addison-Wesley.
8. Effective C++, Scott Meyers, Published by Addison-Wesley.
9. Head First C++ Programming: The Definitive Beginner's Guide, Harry, H. Chaudhary, First Create space Inc, O-D Publishing, LLC USA.
10. Problem Solving with C++, Walter Savitch, Pearson Education.
11. C++ Primer, Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, Published by Addison-Wesley.

Note: *Student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 6*
c. *Output – 4*
d. *Viva- Voce – 5*
e. *Lab Assignment – 3*

DC-MJ-302(TH): Computer System Organization and Architecture

Basic Structure of Computers: Basic Functional Units, Basic Operational Concept, Bus Structure, Performance, Multiprocessor and Multicomputer.
Register Transfer and Micro-operation: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Three State Bus Buffers, memory Transfer, Arithmetic and Logical micro-operations, Shift and Arithmetic shifts.
Basic Computer Organization and Design: Instruction Codes, Stored Program Organization, Indirect Address, Computer Registers, Common Bus System, Computer Instruction, Timing and Control, Instruction Cycle, Register Reference Instructions, Memory Reference Instruction, Input-Output and Interrupt.
CPU Organization: Arithmetic and Logic Unit (ALU) - Combinational ALU, 2's Complement Addition, Subtraction Unit, Booths Algorithm for Multiplication, Division Hardware using Restoration Division Algorithm. General register organization, Accumulator Based, Register Based, Stack Type CPU organization.
Control Unit: Hardwired Control Unit, Micro-programmed Control Unit.
CPU Registers: Program Counter, Stack Pointer Register, Memory Address Register, Instruction Register, Memory Buffer Register, Flag registers, Temporary Registers.
Instructions: Operational Code, Operands, Zero, One, Two and Three Address Instruction, Instruction Types, Addressing modes, Data Transfer and Manipulation instructions, Program control instructions.
CISC and RISC processors: Introduction, relative merits and De-merits.
Input / Output Organization: Polling, Interrupts, subroutines, Memory mapped IO, IO mapped IO, DMA, Bus Arbitration.
Memory: Primary memory: ROM, PROM, EPROM, EEPROM, Flash memory, RAM: SRAM,DRAM, **Cache Memory:** Mapping Functions, Replacement Algorithms, Hit and Miss ratio, Virtual memories, Address Translation, **Secondary Storage:** Magnetic Hard Disks.

Case Study:

Introduction to 8085 Microprocessor: Concepts of Microprocessor and Microcontrollers and their advantages and disadvantages.
Microprocessor Architecture and Memory: Basic Architecture of Microprocessor 8085 and explanation of each block, Addressing modes, Instruction Formats, Instruction Cycle, Clock Cycle, Multiplexed Address Data Bus, Control and Status signals, Microprocessor and Bus Timing, De-multiplexing of Address Data Bus, Generation of Control Signals for I/O and Memory.
Programming 8085: Instruction Set of 8085, Different Programming Techniques, Stack and Subroutines, Counter and Time Delays, Code Conversion, BCD Arithmetic and 16 bit Data Operation.
Interrupts: 8085 Interrupt, RST instructions, Software and Hardware interrupt, multiple Interrupts and Priorities, 8085 Vectored Interrupts, Restart as Software Instruction.

DC-MJ-302(PR): 8085 Microprocessor Lab

Microprocessor 8085 Lab
The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use 8085 Microprocessor Kit to implement.

1. Assembly Language Programming for Arithmetic Operations like Addition, Subtraction, Multiplication and Division on 8, 16 bit data.
2. Assembly Language Programming for different logical operations.
3. Assembly Language Programming for code conversions.
4. Assembly Language Programming for different sorting techniques.
5. Assembly Language Programming for memory block transfer.
6. Assembly Language Programming for AP series and Fibonacci series.
7. Assembly Language Programming for HCF, LCM etc.
8. Assembly Language Programming for Searching.
9. Assembly Language Programming for frequency distribution.
10. Block Replacement and transfer

Text/ Reference Books:

1. Computer System Architecture, Morries Mano, Pearson.
2. Computer Organization & Architecture, Williams Stallings, Pearson.
3. Computer Organization, Hamacher, Vranesic and Zaky, McGraw Hill.
4. Computer Architecture and Organization, Govindrajalu, Tata McGraw Hill.
5. Computer Architecture and Organization, J P Hayes, Tata McGRaw Hill.
6. Structured Computer Organization, Andrew S. Tanenbaum, Austin, Pearson.
7. Microprocessor architecture, programming and applications with 8085/8085A, Wiley eastern Ltd, by Ramesh S. Gaonkar.
8. Intel Corp: The 8085 / 8085A. Microprocessor Book – Intel marketing communication, Wiley inter science publications.
9. Advanced Microprocessors by Ray and Bhurchandi - TMH.
10. Intel Corp. Micro Controller Handbook – Intel Publications.
11. Microprocessors and Interfacing by Douglas V. Hall, McGraw Hill International

Note: *Student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 6*
c. *Output – 4*
d. *Viva- Voce – 5*
e. *Lab Assignment – 3*

Semester – IV

DC-MJ-401(TH): Discrete Mathematics

Number Systems: Introduction to number system, Weighted and Non-Weighted Codes, positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Gray Codes, Alphanumeric codes, ASCII, EBCDIC, Conversion of bases.
Sets: finite and Infinite sets, un-countably Infinite Set; Venn diagrams, set operations; functions- Domain, target, and range/image of a function - Surjections, injections, bijections, Inverses, Composition; relations- Reflexivity, symmetry, anti-symmetry, transitivity, Equivalence relations, partial orders, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion.
Discrete Probability: Finite probability space, events, Properties of events, Conditional probability, Bayes' theorem, Independence.
Recurrences: Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Master Theorem, Growth of Functions: Asymptotic Notations.
Propositional Logic: Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.

Graph Theory: Basic Terminology, Models and Types, Multi-graphs and Weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Trees, Basic Terminology and properties of Trees, Binary Tree, Introduction to Spanning Tree.

Text/ Reference Books:

1. Elements of Discrete mathematics, C.L. Liu , D.P. Mahopatra, Tata McGraw Hill
2. Discrete Mathematics and Its Applications, Kenneth Rosen, McGraw Hill
3. Introduction to algorithms ,T.H. Coremen, C.E. Leiserson, R. L. Rivest, Prentice Hall on India,
4. Discrete Mathematics with Algorithms , M. O. Albertson and J. P. Hutchinson, John wiley Publication,
5. Discrete Structures, Logic, and Computability, J. L. Hein, , Jones and Bartlett Publishers,
6. Essentials of Discrete Mathematics, D.J. Hunter, Jones and Bartlett Publishers
7. Discrete Mathematical Structures with Applications to Combinatorics, V Ramaswamy, University Press
8. Discrete Mathematics: A Concept-based Approach, Basavaraj S Anami, Venkanna S Madalli, University Press
9. An Introduction to Probability Theory and its Applications, Vol. 1 & 2. William Feller. New York, NY: Wiley.
10. Probability Models for Computer Science. Sheldon Ross. San Diego, CA: Harcourt/Academic Press.

DC-MJ-402(TH): Data Structures & Algorithms using C++

Introduction to Data Structure: Abstract Data Type.
Array: 1D, 2D and Multi-dimensional Arrays, Sparse Matrices. Polynomial representation (Polynomial Representation as Application).
Linked List: Singly, Doubly and Circular Lists; Polynomial representation (Polynomial Representation as Application).
Stack: Implementing single / multiple stacks in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack.
Queue: Array and Linked representation of Queue, Circular Queue, De-queue, Priority Queues. Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation).
Tree: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Tree), Binary Search Trees; Height-Balanced Trees (Various operations on AVL Trees).
Searching and Sorting: Linear Search, Binary Search, Comparison of Linear and Binary Search.
Sort: Bubble sort, Selection Sort, Insertion Sort, Merge Sort, Quick sort, Heap Sort, Comparison of Sorting Techniques.
Hashing: Introduction to Hashing, Choosing a Hash Function, collision resolution techniques.
Graph Theory: Graph Representation, Spanning Tree, MST- Kruskal, and Prims algorithm, Reachability test using Warshall algorithm, All pair shortest path - Floyd algorithm, Single source shortest path - Dijkstra Algorithm, Graph Traversal-BFS and DFS algorithm.

DC-MJ-402(PR): Data Structures & Algorithms using C++ Lab

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C++ compiler **(GCC)** for practical.

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search.

2. Write a program using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort, Selection sort etc.
3. Implement Singly Linked List. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists.
4. Implement Doubly Linked List. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation.
8. Perform Queues operations using Circular Array implementation.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. Write a program to scan a polynomial using linked list and add two polynomial.
11. Write a program to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. Write a program to display Fibonacci series (i)using recursion, (ii) using iteration.
13. Write a program to calculate GCD of 2 number (i) with recursion (ii) without recursion.
14. Write a program to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive and Iterative Implementation) (b) Deletion by copying (c) Deletion by Merging (d) Search a no. in BST (e) Display its preorder, postorder and inorder traversals Recursively (f) Display its preorder, postorder and inorder traversals Iteratively (g) Display its level-by-level traversals (h) Count the non-leaf nodes and leaf nodes (i) Display height of tree
(j) Create a mirror image of tree (k) Check whether two BSTs are equal or not
15. Write a program to reverse the order of the elements in the stack using additional stack.
16. Write a program to display the minimum spanning tree using Prim's algorithm.
17. Write a program to display all pair shortest path using Floyd's algorithm.
18. Write a program to display the shortest path using Dijkstra algorithm.

Text/ Reference Books:

1) Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Pr.
2) Data Structures: A Pseudocode Approach with C, Richard F. Gilberg and Behrouz A.Forouzan, Cengage Learning
3) Data Structures In C, Noel Kalicharan, CreateSpace Independent Publishing Platform.
4) Data Structures and algorithm in C, Adam Drozdek, Cengage Learning.
5) The C Programming Language, Brian W. Kernighan and Dennis Ritchie, PrenticeHall.
6) Data Structures Using C and C++, Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, PHI.
7) Classic Data Structures, Debasis Samanta, PHI
8) Fundamental of Computer Algorithms, Horowitz, Sahni, Rajasekaran, Universities Press.
9) Graph Theory with Applications to Engineering & Computer Science. Narsingh Deo, Prentice Hall India
10) Introduction to Graph Theory. Douglas Brent West, Pearson Education India

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 6*
c. *Execution – 4*
d. *Viva- Voce – 5*
e. *Lab Assignment – 3*

DC-MJ-403(TH): Operating System

Introduction: Basic OS functions, types of operating systems: batch systems–multiprogramming systems, time sharing systems;
Operating System Organization: Processor and user modes, kernels, system calls and system programs.
Process: System view of the process and resources, process hierarchy, threads, threading issues.
Process Scheduling: Scheduling criteria, Pre-emptive and non-preemptive scheduling, Long term, short term and medium term, FCFS, SJF, SRTF, Priority scheduling, Round Robin, Multilevel Queue Scheduling, Multilevel Queue Feedback Scheduling.
Process Synchronization: Concurrent Processes, critical section, semaphores and application, methods for inter-process communication;
Deadlock: Definition, Prevention, Avoidance, Detection, Recovery, Banker's algorithm.
Memory Management: Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory
File and I/O Management: Directory structure, file operations, file allocation methods, disc management.

DC-MJ-403(PR): Operating System Lab

Some sample examples/Commands are given below. More problems can be included related to the theory. Use open source system (Debian OS) for practical.

1. Usage of following commands: ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd.
2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, tput (clear, highlight), bc.
4. Write a shell script to check if the number entered at the command line is prime or not.
5. Write a shell script to modify "cal" command to display calendars of the specified months.
6. Write a shell script to modify "cal" command to display calendars of the specified range of months.
7. Write a shell script to accept a login name. If not a valid login name display message – "Entered login name is invalid".
8. Write a shell script to display date in the mm/dd/yy format.
9. Write a shell script to display on the screen sorted output of "who" command along with the total number of users .
10. Write a shell script to display the multiplication table any number,
11. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
12. Write a shell script to find the sum of digits of a given number.
13. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
14. Write a shell script to find the LCD(least common divisor) of two numbers.
15. Write a shell script to perform the tasks of basic calculator.
16. Write a shell script to find the power of a given number.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.
20. Program to show the pyramid of special character "*".

Text/ Reference Books:

1. Operating Systems Concepts, A Silberschatz, P.B. Galvin, G. Gagne, John Wiley Publications.
2. Modern Operating Systems, A.S. Tanenbaum, Pearson Education.
3. Operating Systems: A Modern Perspective, G. Nutt, Pearson Education.
4. Operating Systems, Internals & Design Principles W.Stallings, PHI.
5. Operating Systems- Concepts and design, M. Milenkovic, Tata McGraw Hill.
6. UNIX Concepts and Applications, Sumitabha Das , Tata McGraw-Hill.
7. Understanding the Linux Kernel, D. P. Bovet and M. Cesati, O'Reilly.

Note: *Student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Program – 4*
b. *Implementation – 8*
c. *Viva- Voce – 5*
d. *Lab Assignment – 3*

Semester – V

DC-MJ-501(TH): Theory of Computation and Compiler Design

Finite Automata: Definition of a Finite Automaton, Model, Representation, Classification – with respect to output function Mealy and Moore Machines, State Transition – Deterministic and Non-Deterministic Machine, Examples, Conversion algorithms Mealy to Moore and Moore to Mealy, Non-Deterministic to equivalent Deterministic Finite automata, Finite and Infinite state machines, Removal of Null-transitions, Acceptability of String by a Finite Automaton, Design of different Finite State Machines, Minimized Equivalent Machine.
Formal Languages and Grammar: Introduction to Formal Grammar and Language, Formal Definition, Chomsky's Classification of Grammar – Type 0, Type-1 or Context Sensitive, Type-2 or Context Free and Type-3 or Regular Grammar, Illustration of each of these classes with example, Sentential form, Sentences – Languages or strings, Derivations – left, right derivation, Derivation tree, Parse Tree, Syntax Tree, Ambiguous Grammar and Language, Designing of Grammar for a language, Finding Language for Given Grammar; Definition and basic idea about Push Down Automaton
Regular Expression: Basic Idea and Definition, Regular Expression basic Identities, Arden's Theorem and application for reduction of equivalent regular expressions, Thompson's Construction Algorithm – Regular expression to Finite Automata conversion, State Transition System to Regular Expression conversion algorithm by Arden's Algebraic Method, FA to Regular Grammar and Regular Grammar to FA conversion algorithms and applications.
Turing Machine: Concepts of Turing Machine, Formal Definitions, Classifications – Deterministic and Non-Deterministic Turing Machines, Simple Design of Turing Machines like – Unary Adder, Subtractor, Concatenator, Odd / even count etc and concepts of Universal Turing Machines.
Introduction to Compiler: Overview of compilation, Phases of a compiler
Lexical Analysis: Role of a Lexical analyser, Specification and recognition of tokens, Symbol table, lex.
Parsing: Top down and bottom up parsing (Basics of LR parser).
Intermediate representations: Three address code generation & representations.

Text/ Reference Books:

1. Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, 3rd Edition, Pearson.
2. Theory of Computer Science (Automata, Languages & Computation) by K L P Misra & N Chandrasekharan, PHI.
3. Introduction to Theory of Computation by Micheal Sipser, Cengage Learning.
4. Switching and Finite Automata Theory by Zvi Kohavi, Niraj.K.Jha, TMH.
5. Formal Language and Automata, P. Linz, Narosa
6. Systems Programming, Santanu Chattopadhyaya, PHI.
7. Compilers: Principles, Techniques, and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Prentice Hall.
8. Systems Programming, D. M. Dhamdhere, Tata McGraw Hill.
9. System Software: An Introduction to System Programming, Leland Beck, D. Manjula, Pearson Education.
10. Modern Compiler Design, Grune D, Van Reeuwijk . K, Bal H. E, Jacobs C J H, Langendoen K, Springer.

DC-MJ-502(TH): Computer Graphics using OpenGL

Introduction: Basic concepts of Graphics Devices– Monochrome and Color Monitor displaying technique only, Basic idea for image or picture formation using pixels –Raster Scan and Vector Scan, Image Color Model, Color Coding, Lookup Table based color mapping.
Basic geometrical shapes formation algorithms: Concepts Co-ordinate System, Line Segment, Circle, elliptic segment and its formation; DDA, Bresenham's and Midpoint scan conversion algorithms.
Two and Three Dimensional Transformations: Geometric Transformations operations - Translation, Rotation, Scaling. Reflection, Shearing, Homogeneous coordinate system representation, matrix representation Coordinate Transformations operations - Translation, Rotation, Scaling. Reflection, Shearing, Composite Transformations Operations – Basic ideas and matrix representations by matrix concatenation for a particular operation.
Clipping: Point Clipping, Line Clipping – Region coding, Cohen-Sutherland Algorithm;
Area filling: Boundary fill and flood fill
Projection: Basic Concept of Projection operation and its application, Classification – Perspective, Parallel.
Applications: Basic Concepts Computer Art – publishing, drawing and drafting, Animation – Animating and modelling of real world, Morphing – Classification of morphing and Application to the Advertisements and publicities.

DC-MJ-502(PR): Computer Graphics using OpenGL Lab

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source OpenGL compiler for practical.

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement mid-point circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to apply various 2D transformations on a 2D object (use homogenous coordinates).
5. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.

Text/ Reference Books:

1. Computer Graphics by Zhigang Xiang, Roy Plastock, Schaum's Outlines Series.
2. Computer Graphics by Hern & Baker.
3. Procedural Elements for Computer Graphics by David F. Roger, 2nd Edition, TMH.
4. Computer Graphics by Folly & Vandam.

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 6*
c. *Execution – 4*
d. *Viva- Voce – 5*
e. *Lab Assignment – 3*

DC-MJ-503(TH): Advanced Database Management System

Introduction: Drawbacks of file System; Advantages of DBMS; Layered Architecture of Database, Data Independence; Data Models; Schemas And Instances; Database Languages; Database Users, DBA; Data Dictionary; Functional Components of a DBMS.
Entity Relationship(ER) Modelling: Entity, Attributes and Relationship, Structural Constraints, Keys, ER Diagram of Some Example Database, Weak Entity Set, Specialization and Generalization, Constraints of Specialization and Generalization, Aggregation.
Relational Model: Basic Concepts of Relational Model; Relational Algebra.
Integrity Constraints: Domain Constraints, Referential Integrity, Assertions, Triggers.
Relational Database Design: Problems of Un-Normalized Database; Functional Dependencies (FD), Derivation Rules, Closure of FD Set, Membership of a Dependency, Canonical Cover; Decomposition to 1NF, 2NF, 3NF and BCNF Using FD; Lossless Join Decomposition Algorithm; Dependency preservation.
SQL: Basic Structure, Data Definition, Constraints and Schema Changes; Basic SQL Queries (Selection, Insertion, Deletion, Update); Order by Clause; Complex Queries, Aggregate Function and Group by Clause; Nested Sub Queries; Correlated Sub Queries; Views (Insert-Able and Updatable), Joined Relations; Set Comparisons (All, Some).
Record Storage and File Organization (Concepts only): Fixed Length and Variable Length Records; Spanned and Un-Spanned Organization of Records; Primary File Organizations and Access Structures Concepts; Unordered, Sequential, Hashed; Concepts of Primary and Secondary Index; Dense and Sparse Index; Index Sequential Files; Multilevel Indices.
Transaction Processing (Concepts only): ACID Properties; Transaction States, Concurrent Execution; Serializability (Conflict and View), Recoverability, Test for Serializability.
Introduction to NoSQL & New Data Models: Basics of Key-Value Stores (e.g., Redis), Document Stores (e.g., MongoDB)

DC-MJ-503(PR): Advanced Database Management System Lab

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source MySQL for practical.

Consider the following schema for an Online Retail Database.

CUSTOMER (Customer_id, Name, Email, Phone)
PRODUCT (Product_id, Product_Name, Price, Category)
ORDER (Order_id, Order_Date, Customer_id)

ORDER_DETAILS (Order_id, Product_id, Quantity)

Create the tables maintaining proper primary key, foreign key, and domain constraints. Insert appropriate data for result verification. Write the SQL queries to find the following.

1. Retrieve all customer names and phone numbers.
2. List all products with their price and category.
3. Find all orders placed by a specific customer (e.g., Customer_id = 101).
4. Count the total number of customers in the database.
5. Display the order details for a given order ID.
6. List the names of customers who have placed at least one order.
7. Find the total quantity sold for each product.
8. Show all orders placed in the last 7 days.
9. Identify the most expensive product in each category.
10. Calculate the total cost for each order.
11. Identify the customer who spent the most across all orders.
12. List all products that have never been ordered.
13. Find the average order value for each customer.
14. List the top 3 customers based on the total quantity of products they ordered.
15. Display the number of orders placed on each date.

For MongoDB:

a. Insert sample student records into a MongoDB collection.
b. Query documents with conditions like age > 20 or course = "CS".
c. Update a field (e.g., change a student's address or phone number).
d. Delete documents matching a specific condition.
e. Create an index on the student_id field and check its effect on query speed.
f. Model nested data for orders and items in an e-commerce example.

Text/ Reference Books:

1. Fundamentals of Database Systems 6th Edition, R. Elmasri, S.B. Navathe, Pearson Education.
2. Database Management Systems, R. Ramakrishanan, J. Gehrke,3rd Edition, McGraw-Hill.
3. Database System Concepts 6th Edition, A. Silberschatz, H.F. Korth, S. Sudarshan,McGraw Hill.
4. Database Systems Models, Languages, Design and application Programming, R. Elmasri, S.B. Navathe, Pearson Education.
5. SQL and Relational Theory: How to Write Accurate SQL Code, Christopher J. Date, O'Reilly Media
6. Database Systems: A Practical Approach to Design, Implementation and Management, Thomas M. Connolly and Carolyn E. Begg, Pearson
7. Designing Data-Intensive Applications, Martin Kleppmann, Shroff/O'Reilly

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. Query – 10
b. Proper Data Insertion – 2
c. Viva- Voce – 5
d. Lab Assignment – 3

DC-MJ-504(PR): Programming through Python

Overview of Programming: Structure of a Python Program, Elements of Python
Introduction to Python: Python Interpreter, Using Python as calculator, Python shell, Indentation, Atoms, Identifiers and keywords, Literals, Strings, Operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator)
Creating Python Programs: Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments, Exception handling.
Iteration and Recursion: Conditional execution, Alternative execution, Nested conditionals, Return statement, Recursion, Stack diagrams for recursive functions, Multiple assignment, while statement, for statement.

Strings and Lists: String as a compound data type, Length, Traversal and the for loop, String slices, String comparison, A find function, Looping and counting, List values, Accessing elements, List length, List membership, Lists and for loops, List operations, List deletion; Nested lists.
Object Oriented Programming: Introduction to Classes, Objects and Methods, Standard Libraries.

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source Jupyter Notebook for practical.

1. Running instructions in Interactive interpreter and a Python Script.
2. Write a program to compute distance between two points taking input from the user.
3. Write a program using a for loop that loops over a sequence.
4. Write a program to count the numbers of characters in the string and store them in a dictionary data structure.
5. Write a program to print each line of a file in reverse order.
6. Find mean, median, mode for the given set of numbers in a list using function

Text/ Reference Books:

1. Introduction to Computation and Programming Using Python, John V. Guttag, MIT Press.
2. Think Python: How to Think Like a Computer Scientist, Allen Downey, O'Reilly.
3. Learning Python, Mark Lutz, O'Reilly.
4. Python Programming for the Absolute Beginner, Michael Dawson, Cengage Learning.
5. Learning to Program in Python, P. M. Heathcote, PG Online Limited.
6. Python Programming Fundamentals, Authors: Lee and Kent D.

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. For practical there will be two sections, one for program implementation and another for descriptive question(s). Distribution of marks will be as per the following-*

a. *Program – 20*
b. *Descriptive Question – 10*
c. *Viva- Voce – 15*
d. *Lab Assignment – 5*

Semester – VI

DC-MJ-601(TH): Machine Learning

Machine Learning: Introduction, Machine Learning Process; Machine Learning Types: Supervised, Unsupervised, and Reinforcement Learning; Applications of ML; Difference between AI, and ML;
Supervised Learning: Regression: Linear Regression, Evaluation metrics: MSE, RMSE, MAE, R^2;
Classification: Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes Classifier, Support Vector Machine (SVM), Decision Trees, Ensemble Techniques - Bagging: Random Forest, Boosting: AdaBoost.
Unsupervised Learning: Clustering - K-Means, Hierarchical Clustering, DBSCAN; Dimensionality Reduction: Principal Component Analysis (PCA)
Model Evaluation and Validation: Training, Testing, and Validation Sets; Cross-Validation, Bias-Variance Trade-off, Overfitting and Underfitting; Performance Metrices: Confusion Matrix, Accuracy, Precision, Recall, F1-Score, ROC Curve, MCC, CK, Specificity, Gmean, Youden Index etc.
Reinforcement Learning (Introductory): Basic Concepts: Agent, Environment, Reward, Q-learning

DC-MJ-601(PR): Machine Learning Lab

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source Jupyter Notebook for practical.

1. Write a Python program to demonstrate the basic process of Machine Learning using a small dataset (e.g., Iris or Boston Housing).
2. Prepare a report on applications of Machine Learning in healthcare or finance with examples.
3. Implement Linear Regression on the Boston Housing dataset. Plot the regression line and calculate MSE, MAE, and R^2.
4. Perform Polynomial Regression on a custom dataset and compare it with Linear Regression performance using error metrics.
5. Implement Logistic Regression on the Breast Cancer dataset and evaluate using confusion matrix and classification report.
6. Develop a K-Nearest Neighbors (KNN) classifier on the Iris dataset. Test for different values of 'k' and observe the accuracy.
7. Use the Naive Bayes classifier on a text classification problem (e.g., spam detection).
8. Train and evaluate a Support Vector Machine (SVM) for handwritten digit recognition using the MNIST dataset.
9. Build a Decision Tree classifier and visualize the tree using Scikit-learn on the Titanic dataset.
10. Train a Random Forest classifier and compare its performance with a single Decision Tree on the same dataset.
11. Implement AdaBoost with a base estimator and test its accuracy on a classification problem.
12. Apply K-Means clustering on a dataset and visualize the clusters using a scatter plot.
13. Use Hierarchical Clustering on a dataset and plot the dendrogram.
14. Perform DBSCAN clustering and compare the results with K-Means.
15. Implement Principal Component Analysis (PCA) on the Wine dataset and visualize the first two principal components.
16. Split a dataset into training, validation, and testing sets. Use k-fold cross-validation and report the average accuracy.
17. Demonstrate overfitting and underfitting using a simple ML model by changing the complexity or training data size.
18. Write a program to calculate confusion matrix, precision, recall, F1-score, and ROC curve for a binary classifier.

Module 8: Reinforcement Learning (Introductory)

19. Simulate a Q-learning algorithm for a simple environment (e.g., grid world or taxi problem) using OpenAI Gym.

Text/ Reference Books:

1. Machine Learning, Tom M. Mitchell, McGraw Hill Education
2. Pattern Recognition and Machine Learning, Christopher Bishop, SPRINGER NP EXCLUSIVE
3. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurélien Géron, Shroff/O'Reilly
4. Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Müller and Sarah Guido, Shroff/O'Reilly

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following.*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 6*
c. *Execution – 4*
d. *Viva- Voce – 5*
e. *Lab Assignment – 3*

DC-MJ-602(PR): Data Analytics using Python

Data Acquisition: Reading structured data (CSV, Excel etc.); reading data from JSON and APIs; handling large datasets efficiently using chunking and memory optimization.
Data Cleaning and Preparation: Handling missing values - Mean, Median, Mode, kNN, MI, MICE, EM; detecting and removing duplicates; correcting data types; text cleaning and basic string operations.
Data Transformation and Aggregation: Creating derived variables; filtering and sorting; grouping, summarizing, and aggregating data; reshaping with pivot and melt.
Exploratory Data Analysis (EDA): Generating summary statistics; detecting outliers - IQR, Z Score, Isolation Forest, DBSCAN, Kmean, ECOD; visualizing distributions; identifying correlations-Pearson Correlation.
Data Visualization: Creating basic and advanced plots using Matplotlib and Seaborn - bar plots, histograms, pie charts, line graphs, scatter plots, boxplots, heatmaps, pair plots.
Introduction to Data Mining: Understanding association rules; applying Apriori; interpreting support, confidence, and lift; analyzing frequent itemsets.
Regression: Linear and Polynomial Regression
Classification: KNN, Naïve Bayes, Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, eXtreame Gradient Boosting, Extra Trees.
Clustering: Understanding distance metrics- Minkowski-Manhattan, Euclidian, Hamming; K-Means clustering; evaluating clustering output; visualizing clusters with PCA, t-SNE, UMAP (2D/3D).
Mini Project / Case Study: Full-cycle data analysis on a real-world dataset (e.g., social, healthcare, sales); includes cleaning, EDA, visualization, and basic modeling or pattern discovery.

Tools and Libraries — Platform: Jupyter Notebook, Language: Python 3.x, Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, mlxtend, json, requests etc.

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source Jupyter Notebook for practical.

Data Acquisition
Load a CSV file containing COVID-19 daily case numbers. Display the first 5 rows and the column names.
Fetch JSON data from a public API (e.g., OpenWeatherMap, COVID19API) and convert it into a Pandas DataFrame.
Read an Excel file with multiple sheets. Load data from each sheet and print summary statistics.
Data Cleaning and Preparation
Load a dataset with missing values. Use appropriate techniques to fill or drop missing entries.
Detect and remove duplicate rows from a dataset.
Convert a column of dates (as strings) to Python datetime objects and extract year and month.
Clean a column containing user-entered product names with inconsistent casing and extra spaces.
Data Transformation and Aggregation
Create a new column in a dataset representing the profit = sales - cost.
Group a sales dataset by region and calculate total and average sales.
Use pivot_table to summarize sales data by month and product category.
Reshape a dataset using melt to convert wide data into long format.
Exploratory Data Analysis (EDA)
Generate summary statistics (mean, median, mode, std) for all numeric columns.
Identify outliers in a dataset using the IQR method.
Compute and display the correlation matrix for a dataset.
Data Visualization
Plot a histogram and boxplot for the “age” column from a dataset.
Create a line plot of monthly sales trend using matplotlib.
Use seaborn.pairplot() to visualize relationships between numeric features.
Create a heatmap to display correlations between variables in a dataset.
Clustering
Perform K-Means clustering on an IRIS dataset. Visualize the clusters using PCA.
Plot the elbow curve to determine the optimal number of clusters.
Visualize clustered points using scatter plot with cluster labels.
Introduction to Data Mining
Load a grocery transactions dataset and apply the Apriori algorithm to find frequent itemsets.
Generate association rules from frequent itemsets and filter by lift > 1.
Interpret one or two strong association rules in natural language.
Mini Project / Case Study
Perform full-cycle analysis on a real-world dataset (e.g., Titanic, COVID, sales):
- Load and clean the data
- Perform EDA and visualize key insights

- Apply clustering or association rule mining as applicable
- Present findings in a summarized report (tables and plots)

Text/ Reference Books:

1. Python for Data Analysis, Wes McKinney, O'Reilly.
2. Hands-On Data Analysis with Pandas, Stefanie Molin, Packt.
3. Data Mining: Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann.
4. Practical Statistics for Data Scientists, Peter Bruce, Andrew Bruce, Peter Gedeck, O'Reilly.
5. Effective Data Storytelling, Brent Dykes, Wiley.
6. Python Data Science Handbook, Jake VanderPlas, O'Reilly.

Note: *For Practical, students can use library functions to solve the problem. For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following.*

a. *Algorithm / Flowchart – 5*
b. *Program Execution – 25*
c. *Viva- Voce – 15*
d. *Lab Assignment-5*

DC-MJ-603(TH): Data Communication and Networking

Data Communication Concepts: Analog and Digital Signals, Periodic and Non-periodic signals, Time and Frequency Domain, Bandwidth and Data rate, Signal rate, Serial and Parallel Transmission. Protocol.
Various modes of transmission: Simplex/ Half Duplex, Duplex; Features of guided and unguided transmission media; Circuit Switching, Packet Switching; transmission impairment.
Physical structure of Network: Types of connections (Topologies), Categories of Computer Network: LAN, MAN, WAN;
Digital to Digital conversion: Line coding schemes; Analog to Digital Conversion: PCM, DM; Digital to Analog conversion: ASK, PSK, FSK, QAM; Modulation and Encoding: AM, FM, PM;
Multiplexing: FDM, TDM, WDM; OSI &TCP/IP Model.
Error detection and correction: CRC, Checksum, Hamming Code (SBEDC);
Protocols: IP, ARP, RARP, TCP, UDP, SMTP, FTP, DNS, DHCP etc.

Text/Reference books:

1. Data Communications and Networking ,B. A. Forouzan, THM.
2. Computer Networks, A.S. Tanenbaum, PHI.
3. Data and Computer Communication, W. Stallings, PHI/ Pearson Education
4. Data & Computer Communication, Black, PHI.

DC-MJ-604(PR): Minor Project

Guidelines: Each student of B.Sc. SEM - VI will carry out one project work under the supervision of a faculty member of the college. The project will be assigned at the beginning of SEM – V / VI academic session. The student will submit a project report representing the actual work in a suitable format. The student should defend the project before the examiners. The project work will be evaluated on the basis of presentation and viva-voce examination. The examination will be as per University guidelines.

Project Report should contain the following:

1 Title of the Project
2 Introduction
3 Objectives of the Project
4 Literature Review
5 Methodology
6 Result and Analysis Report in a suitable format
7 Testing and Analysis
8 Conclusion and future scope for development
9 Bibliography

Broad Areas: Real world problem solving using machine learning, Mobile app development, IOT based system, Natural Language processing etc. Following are some examples.

1. Predictive models, classification systems, etc. – Credit Risk Prediction using ML, Breast Cancer Early Detection, Customer Churn Prediction (with Explainable AI)
2. Insights from large or complex datasets – Covid-19 Data Dashboard (Python + Plotly)
3. Sales Data Analysis using Power BI or Tableau – Student Performance Analysis (Jupyter + Pandas).
4. Projects involving sensors and microcontrollers – Smart Dustbin using Arduino + Ultrasonic Sensor, Home Automation with NodeMCU + Firebase, IoT-based Temperature Logger to Google Sheets
5. Android/iOS apps using Kotlin, Java, or Flutter – Location-based To-Do App, Expense Tracker Mobile App, Campus Guide with Google Maps API
6. Text analytics, summarization, chatbots – Resume Ranker using NLP, Fake News Detection, Chatbot for College Enquiry

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following.*

a. *Problem Definition – 10*
b. *Project Documentation – 10*
c. *Project Presentation – 10*
d. *Viva- Voce – 20*

Computer Science Honours

Semester – VII

DC-MJ-701 (TH): Advanced Design and Analysis of Algorithms

Introduction- Fundamentals of Algorithms and Analysis of Algorithms - Orders of Magnitude (Asymptotic notations); Growth rates, some common bounds (constant, logarithmic, linear, polynomial, exponential);
Average and worst case analysis; Analysing control statements; Recurrence Relations- substitution, change of variables, master's method
Divide and conquer algorithms - Introduction - Quick sort, worst and average case complexity, Merge sort, Strassen's Matrix multiplication, Binary search, Finding the maximum and minimum etc.
Greedy algorithms - General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm-Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Shortest path (Dijkstra algorithm), Huffman coding
Dynamic programming – Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming- Fibonacci numbers, Warshall and Floyd algorithm; Matrix chain multiplication, Longest Common Subsequence (LCS), Optimal Binary Search Tree.
Graph Algorithms: - An introduction using graphs - Traversing Trees: Depth First Search, Breath First Search.
Backtracking and Branch and Bound:– 0/1 Knapsack Problem, The Eight queens problem, Travelling Salesman problem.
String matching – Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, KMP Algorithm
Introduction to Complexity Theory - The class P and NP, Polynomial reduction, NP- Complete Problems, NP-Hard Problems.

DC-MJ-701 (PR): Advanced Design and Analysis of Algorithms Lab

Programming must be done using Python preferably using Jupyter Notebook platform. Students will complete assignments involving the implementation and analysis of algorithms and conduct empirical studies corresponding to the topics covered in the theory component of the course.
The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

1. Implementation and Time analysis of sorting algorithms. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort .
2. Implementation and Time analysis of linear and binary search algorithm.
3. Write a program for Strassen's Matrix Multiplication.
4. To implement Huffman coding and analyse its time complexity
5. Write a program for travelling salesman problem.
6. Implementation of chain matrix multiplication using dynamic programming.
7. Implementation of making a change problem using dynamic programming
8. Implementation of a knapsack problem using greedy algorithm
9. Implementation of Graph and Searching (DFS and BFS).
10. Implement prim's algorithm.
11. Implement Kruskal's algorithm.
12. Implement LCS problem.
13. To implement following string matching algorithms and analyse time complexities:
a. Naïve, b. Rabin Karp, c. Knuth Morris Pratt
14. Write a program for Floyd-Warshall algorithm.

Text/ Reference Books:

1. Introduction to Algorithms: Cormen, Leiserson, Rivest and Stein: Prentice Hall of India
2. Fundamentals of Computer Algorithms: Sahni , Horowitz: Universites Press
3. Introduction to the Design and Analysis of Algorithms: AnanyLevitin
4. Data Structures and Algorithms: Aho, Hopcroft and Ullmann: Addison Wesley.
5. Data Structures and Algorithms in Java b: Michael T. Goodrich , Roberto Tamassia
6. Data Structures and Algorithms in C++ : Adam Drozdek
7. Teofilo F. Gonzalez, Handbook of NP-Completeness: Theory and Applications

Note: *Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*
a. *Algorithm / Flowchart - 3*
b. Program/Source Code – 6
c. Viva- Voce – 4
d. Lab Assignment – 2

DC-MJ-702 (TH): Software Engineering

Introduction: Software Engineering – A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile – Agility Principles, Agile Model – Scrum.

SRS: Software Requirements Analysis and Specifications: Use Case Approach, Software Requirement Specification Document, Flow oriented Modelling, Data Flow Modelling, Sequence Diagrams.

Design Modelling: Translating the Requirements model into the Design Model, The Design Process, Design Concepts – Abstraction, Modularity and Functional Independence; Architectural Mapping using Data Flow.

Software Metrics and Project Estimations: Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations, COCOMO II Model); Project Scheduling (Timeline charts, tracking the schedule).

Quality Control and Risk Management: Quality Control and Quality Assurance, Software Process Assessment and Improvement Capability Maturity Model Integration (CMMI); Software Risks, Risk Identification, Risk Projection and Risk Refinement, Risk Mitigation, Monitoring and Management.

Software Testing: Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing, etc.

Text/ Reference Books:

1. Roger S.Pressman, Software engineering- A practitioner's Approach, McGraw-Hill International Editions
2. Ian Sommerville, Software engineering, Pearson education Asia
3. Pankaj Jalote, Software Engineering – A Precise Approach Wiley
4. Software Engineering Fundamentals by Ali Behhforoz & Frederick Hudson OXFORD
5. Rajib Mall, Fundamentals of software Engineering, Prentice Hall of India.
6. Engineering Software as a Service An Agile Software Approach, Armando Fox and David Patterson
7. John M Nicolas, Project Management for Business, Engineering and Technology, Elsev

---

DC-MJ-703(TH): Artificial Intelligence

Introduction: Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behaviour and environment.

Problem Solving and Searching Techniques: Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics

Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.

Knowledge Representation: Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG)

Dealing with Uncertainty and Inconsistencies: Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

Text/ Reference Books:

1. Elaine Rich and Kelvin Knight: Artificial Intelligence, Tata McGraw Hill, 2002
2. Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach (2nd ed.), Pearson Education,2006.
3. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems: Prentice Hall of India, 2006.
4. Nils J Nilson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publishers, Inc., San Francisco, California, 2000.
5. R. Akerkar: Introduction to Artificial Intelligence, Prentice-Hall of India, 2005
6. Nils J. Nilson: Principles of Artificial Intelligence, Narosa Publishing House, 2001
7. W.F. Clocksin and C.S. Mellish: Programming in PROLOG, Narosa Publishing House, 2002.
8. Saroj Kaushik: Logic and Prolog Programming, New Age International Publisher, 2006

Paper: DC-MJ-704 (PR): Research Methodology

Introduction to Research: Meaning and objectives of research; types of research (basic, applied, qualitative, and quantitative); characteristics of good research; research process and steps in conducting research.
Research Problem and Research Design: Identifying and defining a research problem; formulation of research questions and hypotheses; research design and planning; exploratory, descriptive, and experimental research.
Literature Review: Purpose of literature review; sources of research literature; searching academic databases; organizing and summarizing related work; identifying research gaps.
Data Collection Methods: Primary and secondary data; surveys, questionnaires, interviews, and observations; experimental data collection; sampling techniques and sample size considerations.
Data Analysis and Interpretation: Introduction to data analysis; descriptive statistics; basic statistical tools; visualization and interpretation of research data.
Research Tools and Software: Introduction to research tools and software for data analysis and documentation (e.g., spreadsheets, statistical tools, reference management software).
Writing Research Reports and Papers: Structure of research papers; writing abstracts, introductions, methodology, results, and conclusions; preparing technical reports using LaTeX.
Citation, Referencing, Plagiarism and Application of AI tools in Research: Citation styles; referencing techniques; avoiding plagiarism; use of plagiarism detection tools; Different AI tools for research and their limitation.
Presentation of Research: Preparing research presentations; scientific posters; communicating research findings effectively.
Ethics in Research: Ethical issues in research; authorship, data integrity, responsible publication, and academic honesty.

Text/ Reference Books:

1. Research Methodology: Methods and Techniques – C. R. Kothari and Gaurav Garg, New Age International.
2. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches – John W. Creswell, SAGE Publications.
3. The Craft of Research – Wayne C. Booth, Gregory G. Colomb, and Joseph M. Williams, University of Chicago Press.
4. Doing Research in the Real World – David E. Gray, SAGE Publications.
5. Writing for Computer Science – Justin Zobel, Springer.

Note: *This paper is designed to provide hands-on training in research practices. Students will actively perform the different stages of research. Each student will complete practical exercises, prepare documents, and present their work as part of continuous assessment. For documentation and presentation student must use LaTeX as tool. Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp.*

Each student must submit a research project file containing the following using LaTeX:

1. *Research problem and objectives*
2. *Literature review*
3. *Data collection method*
4. *Data analysis results*
5. *Research report*
6. *References and citations*
7. *Presentation slides*

Distribution of marks will be as per the following-

a. *Practical Evaluation – 25*
b. *Viva- Voce – 15*
c. *Lab Assignment- 10*

Semester – VIII

Paper: DC-MJ-801 (TH): Deep Learning

Introduction to Deep Learning: Overview of artificial neural networks, perceptron and multi-layer perceptrons (MLP), activation functions, representation learning, and the evolution of deep learning.
Backpropagation and Optimization: Backpropagation algorithm, gradient descent, loss functions, learning rate scheduling, and optimization algorithms such as Adam and RMSProp.
Datasets, Learning and Regularization: Training, validation, and test datasets; overfitting and underfitting; regularization methods such as dropout, batch normalization, and early stopping.
Training Deep Neural Networks: Weight initialization, optimization challenges, vanishing and exploding gradient problems, and strategies for training deep models.
Convolutional Neural Networks (CNNs): Convolution operations, pooling, padding, feature extraction, and State-of-the-art CNN architectures.
Recurrent Neural Networks (RNNs) and Sequence Modeling: Sequence learning, RNN, LSTM, and GRU; their applications in text processing, speech recognition, and time-series analysis.
Transformer Architecture and Attention Mechanisms: Self-attention, encoder-decoder architecture, positional encoding, and Transformer models for sequence learning.
Generative Deep Learning: Introduction to generative models including Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion models.
Large Language Models (LLMs): Foundations of LLMs, pre-training, fine-tuning, prompt engineering, and applications in natural language understanding and generation.
Vision-Language Models and Multimodal AI: Large Vision-Language Models (LVLMs), multimodal learning, image-text alignment models, and multimodal reasoning.

Paper: DC-MJ-801 (PR): Deep Learning Lab

Implement using Python and suitable deep learning frameworks. Students will complete assignments involving the implementation, training, and evaluation of neural networks, convolutional neural networks, recurrent neural networks, transformer models, and other deep learning architectures related to the topics covered in the theory component of the course. Empirical studies on model performance using standard datasets will also be conducted. The practical work will also explore applications of deep learning in areas such as computer vision, speech recognition, natural language processing, healthcare, robotics, multimedia analysis etc.

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

1. Write a Python program to implement a Multi-Layer Perceptron (MLP) for classification using any standard dataset (e.g., Iris or MNIST).
   - Use at least one hidden layer.
   - Train the model and report accuracy and loss.
2. Write a Python program to implement and visualize the following activation functions:
   - Sigmoid
   - ReLU
   - Tanh

   Plot their graphs and briefly compare their properties.
3. Implement a simple neural network using backpropagation to classify a dataset (e.g., Iris dataset).
   Display:
   - Training loss
   - Accuracy
4. Design and train a Convolutional Neural Network (CNN) using TensorFlow/Keras for image classification on the MNIST dataset.
   Tasks:
   - Define CNN layers
   - Train the model
   - Display test accuracy
5. Train a neural network model and demonstrate overfitting. Then apply one of the following techniques:
   - Dropout
   - Batch Normalization
   - Early stopping

   Compare model performance before and after regularization.

6. Implement an RNN or LSTM model to predict the next value in a simple time-series dataset (e.g., stock prices or sine wave data).
7. Write a Python program to demonstrate a simple attention mechanism using matrix operations and visualize the attention weights.
8. Implement a simple Generative Adversarial Network (GAN) or Variational Autoencoder (VAE) to generate images using the MNIST dataset.
Display:

- Generated samples
- Training loss

9. Use a dataset of text (for example movie reviews) and build a text classification model using an RNN/LSTM.
Tasks:

- Preprocess text data
- Train the model
- Evaluate accuracy.

10. Use a pre-trained transformer model (such as BERT) using HuggingFace to perform one of the following tasks:

- Sentiment analysis
- Text summarization
- Question answering

Display the output for sample inputs.

Text/ Reference Books:

1. Deep Learning – Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press.
2. Pattern Recognition and Machine Learning – Christopher M. Bishop, Springer.
3. Neural Networks and Deep Learning – Michael Nielsen, Determination Press.
4. Deep Learning with Python – François Chollet, Manning Publications.
5. Speech and Language Processing – Daniel Jurafsky and James H. Martin, Pearson.
6. Dive into Deep Learning – Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Cambridge University Press.
7. Generative Deep Learning – David Foster, O'Reilly Media.
8. Natural Language Processing with Transformers – Lewis Tunstall, Leandro von Werra, and Thomas Wolf, O'Reilly Media.

Note: *Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Implementation and execution - 8*
b. Viva- Voce – 5
c. Lab Assignment – 2

Paper: DC-MJ-802 (TH): Computational Intelligence

Introduction to Computational Intelligence, Computational Intelligence vs Artificial Intelligence.
Rough Sets: Introduction, Set Approximation, Decision Tables.

Fuzzy Logic Systems: Notion of fuzziness, fuzzy modeling, operations on fuzzy sets, T-norms and other aggregation operators, basics of approximate reasoning, compositional rule of inference, fuzzy rule-based systems, (Takagi-Sugeno and Mamdani-Assilian models), schemes of fuzzification, inferencing, defuzzificatin, fuzzy clustering, fuzzy rule-based classifier.

Artificial Neural Networks: The neuron as a simple computing element the Perceptron, Multilayer Neural Networks, Supervised Learning Neural Networks, Unsupervised Learning Neural Networks, Radial Basis Function Networks, Reinforcement Learning

Evolutionary Computation: Genetic operators, building block hypothesis, evolution of structure, genetic algorithms based on tree and linear graphs, applications in science and engineering; Particle Swarm Optimization.

Books and References:

1. Leszek Rutkowski, Computational Intelligence: Methods and Techniques, Springer,2008.
2. Andries P.Engelbrecht, Computational Intelligence: An Introduction, John Wiley and Sons, 2007.
3. K. H. Lee, First Course on Fuzzy Theory and Applications, Springer, 2005
4. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, 1989
5. E. Alpaydin, Introduction to Machine Learning, Prentice-Hall of India, 2004
6. Amit Konar, Computational Intelligence: Principles, Techniques and Applications, Springer, 2005.

Paper: DC-MJ-803 (PR): Web Programming

Web Essentials: Clients, Servers and Communication; The Internet; Basic Internet protocols; The WWW; HTTP request message and response message; Web clients and web servers – case study.

Introduction to HTML: HTML, HTML domains, basic structure of an HTML document, creating an HTML document, markup tags, heading, paragraphs, line breaks, HTML tags. Elements of HTML, working with text, lists, tables and frames, working with hyperlinks, images and multimedia, forms and controls.

Introduction to Cascading Style Sheets: Concepts of CSS, creating style sheets, CSS properties, CSS styling (background, text format, controlling fonts), working with block elements and objects. Working with lists and tables, CSS ID and class. Box model (introduction, border properties, padding properties, margin properties), CSS colour, grouping, dimensions, display, positioning, floating, align, pseudo class, navigation bar, image sprites.

JavaScript: Client-side scripting, introduction to JavaScript, simple JavaScript, variables, functions, conditions, loops and repetitions. JavaScript and objects, JavaScript own objects, the DOM and web browser environment, forms and validations.

DHTML: Combining HTML, CSS and JavaScript, events and buttons, controlling the browser.

PHP: Starting to script on server side, PHP basics, variables, data types, operators, expressions, constants, decisions and loops for making decisions. Strings – creating, accessing, searching, replacing and formatting strings. Arrays – creation, accessing arrays, multidimensional arrays. PHP with database.

Some sample lab examples are given below. More problems can be included.

1. Acquaintance with elements, tags and basic structure of HTML files.
2. Practicing basic and advanced text formatting.
3. Practice use of image, video and sound in HTML documents.
4. Designing of web pages – document layout, lists, tables.
5. Practicing hyperlinks of web pages, working with frames.
6. Working with forms and controls.
7. Acquaintance with creating style sheets, CSS properties and styling.
8. Working with background, text, font and list properties.
9. Working with HTML element box properties in CSS.
10. Develop a simple calculator for addition, subtraction, multiplication and division using JavaScript.
11. Create an HTML page with JavaScript that takes an integer number as input and tells whether the number is odd or even. Create an HTML page that contains a form with fields name, email, mobile number, gender, favourite colour and a button;
12. Write JavaScript code to validate each entry. Also write code to combine and display the information in a text box when the button is clicked.
13. Write a PHP program to check if a number is prime or not.
14. Write a PHP program to print the first ten Fibonacci numbers.
15. Create a MySQL database and connect with PHP.
16. Write PHP script for storing and retrieving user information from a MySQL table.

a. Write an HTML page which takes name, address, email and mobile number from the user (register.php).
b. Store this data in a MySQL database.
c. Next page displays all users in an HTML table using PHP (display.php).

17. Using HTML, CSS, JavaScript, PHP and MySQL, design an authentication module of a web page.

Text / Reference Books:

1. Web Technologies – Black Book – DreamTech Press
2. Matt Doyle, Beginning PHP 5.3 (Wrox–Wiley Publishing)
3. John Duckett, Beginning HTML, XHTML, CSS and JavaScript
4. HTML, XHTML and CSS Bible, Wiley India – Steven M. Schafer

Note: *Students have to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of Marks as follows.*
a. Program – 25
b. Viva-Voce – 20
c. Lab Assignment – 5

Paper: DC-MJ-804 (PR): Major Project

Major Project : Total credit=4; This should be spilt into two parts: Preparation (03 credits) and presentation (01 credits);

In preparation section, a student will have to focus on a relevant topic (1) *Problem Definition* (2) literature review, (3) research gap, (4) Objective(s), (5) Methodology, (6) Results, (7) Discussion (8) Conclusion and Future Scope and (9) References (following internationally accepted style)

Guidelines: Each student of B.Sc. SEM - VIII will carry out one Major Project work under the supervision of a faculty member. The student will submit a project report representing the actual work in a suitable format. The student should defend the project before the examiners. The project work will be evaluated on the basis of presentation and viva-voce examination. The examination will be as per University guidelines.

Broad Areas: Real world problem solving using machine learning, Mobile app development, IOT based system, Natural Language processing etc. Following are some examples.

1. Predictive models, classification systems, etc. – Credit Risk Prediction using ML, Breast Cancer Early Detection, Customer Churn Prediction (with Explainable AI)
2. Insights from large or complex datasets – Covid-19 Data Dashboard (Python + Plotly)
3. Sales Data Analysis using Power BI or Tableau – Student Performance Analysis (Jupyter + Pandas).
4. Projects involving sensors and microcontrollers – Smart Dustbin using Arduino + Ultrasonic Sensor, Home Automation with NodeMCU + Firebase, IoT-based Temperature Logger to Google Sheets
5. Android/iOS apps using Kotlin, Java, or Flutter – Location-based To-Do App, Expense Tracker Mobile App, Campus Guide with Google Maps API
6. Text analytics, summarization, chatbots – Resume Ranker using NLP, Fake News Detection, Chatbot for College Enquiry

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following.*

a. *(1) Problem Definition (Problem Definition, Literature Review, Research Gap): 15*
   (2) Implementation (Objective(S), Methodology, Results, Discussion, Conclusion and Future Scope): 25
   (3) Report (References and Report Writing): 10
b. *Project Presentation – 10*
c. *Viva- Voce – 15*

Computer Science Honours with Research

Semester – VII

DC-MJ-701 (TH): Advanced Design and Analysis of Algorithms

Introduction- Fundamentals of Algorithms and Analysis of Algorithms - Orders of Magnitude (Asymptotic notations); Growth rates, some common bounds (constant, logarithmic, linear, polynomial, exponential); Average and worst case analysis; Analysing control statements; Recurrence Relations- substitution, change of variables, master's method
Divide and conquer algorithms - Introduction - Quick sort, worst and average case complexity, Merge sort, Strassen's Matrix multiplication, Binary search, Finding the maximum and minimum etc.
Greedy algorithms - General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm-Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Shortest path (Dijkstra algorithm), Huffman coding;
Dynamic programming – Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming- Fibonacci numbers, Warshall and Floyd algorithm; Matrix chain multiplication, Longest Common Subsequence (LCS), Optimal Binary Search Tree.
Graph Algorithms: - An introduction using graphs and games - Traversing Trees: Depth First Search, Breath First Search.
Backtracking and Branch and Bound:– 0/1 Knapsack Problem, The Eight queens problem, Travelling Salesman problem.
String matching – Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, KMP Algorithm
Introduction to Complexity Theory - The class P and NP, Polynomial reduction, NP- Complete Problems, NP-Hard Problems.

DC-MJ-701 (PR): Advanced Design and Analysis of Algorithms Lab.

Programming must be done using Python preferably using Jupyter Notebook. Students will complete assignments involving the implementation and analysis of algorithms and conduct empirical studies corresponding to the topics covered in the theory component of the course.

The questions *listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.*

1. Implementation and Time analysis of sorting algorithms. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort .
2. Implementation and Time analysis of linear and binary search algorithm.
3. Write a program for Strassen's Matrix Multiplication.
4. To implement Huffman coding and analyse its time complexity
5. Write a program for travelling salesman problem.
6. Implementation of chain matrix multiplication using dynamic programming.
7. Implementation of making a change problem using dynamic programming
8. Implementation of a knapsack problem using greedy algorithm
9. Implementation of Graph and Searching (DFS and BFS).
10. Implement prim's algorithm.
11. Implement Kruskal's algorithm.
12. Implement LCS problem.
13. To implement following string matching algorithms and analyse time complexities:
a. Naïve, b. Rabin Karp, c. Knuth Morris Pratt
14. Write a program for Floyd-Warshall algorithm.

Text/ Reference Books:

1. Introduction to Algorithms: Cormen, Leiserson, Rivest and Stein: Prentice Hall of India
2. Fundamentals of Computer Algorithms: Sahni , Horowitz: Universites Press
3. Introduction to the Design and Analysis of Algorithms: AnanyLevitin
4. Data Structures and Algorithms: Aho, Hopcroft and Ullmann: Addison Wesley.
5. Data Structures and Algorithms in Java b: Michael T. Goodrich , Roberto Tamassia
6. Data Structures and Algorithms in C++ : Adam Drozdek
7. Teofilo F. Gonzalez, Handbook of NP-Completeness: Theory and Applications

Note: *Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*
a. *Algorithm / Flowchart - 3*
b. Program/Source Code – 6
c. Viva- Voce – 4
d. Lab Assignment – 2

DC-MJ-702 (TH): Software Engineering

Introduction: Software Engineering – A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile – Agility Principles, Agile Model – Scrum.

SRS: Software Requirements Analysis and Specifications: Use Case Approach, Software Requirement Specification Document, Flow oriented Modeling, Data Flow Modeling, Sequence Diagrams.

Design Modeling: Translating the Requirements model into the Design Model, The Design Process, Design Concepts – Abstraction, Modularity and Functional Independence; Architectural Mapping using Data Flow.

Software Metrics and Project Estimations: Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations, COCOMO II Model); Project Scheduling (Timeline charts, tracking the schedule).

Quality Control and Risk Management: Quality Control and Quality Assurance, Software Process Assessment and Improvement Capability Maturity Model Integration (CMMI); Software Risks, Risk Identification, Risk Projection and Risk Refinement, Risk Mitigation, Monitoring and Management.

Software Testing: Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing.

Text/ Reference Books:

1. Roger S.Pressman, Software engineering- A practitioner's Approach, McGraw-Hill International Editions
2. Ian Sommerville, Software engineering, Pearson education Asia
3. Pankaj Jalote, Software Engineering – A Precise Approach Wiley
4. Software Engineering Fundamentals by Ali Behhforoz & Frederick Hudson OXFORD
5. Rajib Mall, Fundamentals of software Engineering, Prentice Hall of India.
6. Engineering Software as a Service An Agile Software Approach, Armando Fox and David Patterson
7. John M Nicolas, Project Management for Business, Engineering and Technology, Elsev

---

DC-MJ-703(TH): Artificial Intelligence

Introduction: Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.

Problem Solving and Searching Techniques: Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics
Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.
Knowledge Representation: Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG)
Dealing with Uncertainty and Inconsistencies: Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

Text/ Reference Books:

1. Elaine Rich and Kelvin Knight: Artificial Intelligence, Tata McGraw Hill, 2002
2. Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach (2nd ed.), Pearson Education,2006.
3. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems: Prentice Hall of India, 2006.
4. Nils J Nilson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publishers, Inc., San Francisco, California, 2000.
5. R. Akerkar: Introduction to Artificial Intelligence, Prentice-Hall of India, 2005
6. Nils J. Nilson: Principles of Artificial Intelligence, Narosa Publishing House, 2001
7. W.F. Clocksin and C.S. Mellish: Programming in PROLOG, Narosa Publishing House, 2002.
8. Saroj Kaushik: Logic and Prolog Programming, New Age International Publisher, 2006

Paper: DC-MJ-704 (PR): Research Methodology and Application

Introduction to Research: Meaning and objectives of research; types of research (basic, applied, qualitative, and quantitative); characteristics of good research; research process and steps in conducting research.
Research Problem and Research Design: Identifying and defining a research problem; formulation of research questions and hypotheses; research design and planning; exploratory, descriptive, and experimental research.
Literature Review: Purpose of literature review; sources of research literature; searching academic databases; organizing and summarizing related work; identifying research gaps.
Data Collection Methods: Primary and secondary data; surveys, questionnaires, interviews, and observations; experimental data collection; sampling techniques and sample size considerations.
Data Analysis and Interpretation: Introduction to data analysis; descriptive statistics; basic statistical tools; visualization and interpretation of research data.
Research Tools and Software: Introduction to research tools and software for data analysis and documentation (e.g., spreadsheets, statistical tools, reference management software).
Writing Research Reports and Papers: Structure of research papers; writing abstracts, introductions, methodology, results, and conclusions; preparing technical reports using LaTeX.
Citation, Referencing, Plagiarism and Application of AI tools in Research: Citation styles; referencing techniques; avoiding plagiarism; use of plagiarism detection tools; Different AI tools for research and their limitation.
Presentation of Research: Preparing research presentations; scientific posters; communicating research findings effectively.
Ethics in Research: Ethical issues in research; authorship, data integrity, responsible publication, and academic honesty.

Text/ Reference Books:

1. Research Methodology: Methods and Techniques – C. R. Kothari and Gaurav Garg, New Age International.
2. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches – John W. Creswell, SAGE Publications.
3. The Craft of Research – Wayne C. Booth, Gregory G. Colomb, and Joseph M. Williams, University of Chicago Press.
4. Doing Research in the Real World – David E. Gray, SAGE Publications.
5. Writing for Computer Science – Justin Zobel, Springer.

Note: *This paper is designed to provide hands-on training in research practices. Students will actively perform the different stages of research. Each student will complete practical exercises, prepare documents, and present their work as part of continuous assessment. For documentation and presentation student must use LaTeX as tool. Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp.*

Each student must submit a research project file containing the following using LaTeX:

1. Research problem and objectives
2. Literature review
3. Data collection method
4. Data analysis results
5. Research report
6. References and citations
7. Presentation slides

Distribution of marks will be as per the following-

a. *Practical Evaluation – 25*
b. *Viva- Voce – 15*
c. *Lab Assignment- 10*

Semester – VIII

Paper: DC-MJ-801 (TH): Deep Learning

Introduction to Deep Learning: Overview of artificial neural networks, perceptron and multi-layer perceptrons (MLP), activation functions, representation learning, and the evolution of deep learning.
Backpropagation and Optimization: Backpropagation algorithm, gradient descent, loss functions, learning rate scheduling, and optimization algorithms such as Adam and RMSProp.
Datasets, Learning and Regularization: Training, validation, and test datasets; overfitting and underfitting; regularization methods such as dropout, batch normalization, and early stopping.
Training Deep Neural Networks: Weight initialization, optimization challenges, vanishing and exploding gradient problems, and strategies for training deep models.
Convolutional Neural Networks (CNNs): Convolution operations, pooling, padding, feature extraction, and State-of-the-art CNN architectures.
Recurrent Neural Networks (RNNs) and Sequence Modelling: Sequence learning, RNN, LSTM, and GRU; their applications in text processing, speech recognition, and time-series analysis.
Transformer Architecture and Attention Mechanisms: Self-attention, encoder-decoder architecture, positional encoding, and Transformer models for sequence learning.
Generative Deep Learning: Introduction to generative models including Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion models.
Large Language Models (LLMs): Foundations of LLMs, pre-training, fine-tuning, prompt engineering, and applications in natural language understanding and generation.
Vision-Language Models and Multimodal AI: Large Vision-Language Models (LVLMs), multimodal learning, image-text alignment models, and multimodal reasoning.

Paper: DC-MJ-801 (PR): Deep Learning Lab

Implement using Python and suitable deep learning frameworks. Students will complete assignments involving the implementation, training, and evaluation of neural networks, convolutional neural networks, recurrent neural networks, transformer models, and other deep learning architectures related to the topics covered in the theory component of the course. Empirical studies on model performance using standard datasets will also be conducted. The practical work will also explore applications of deep learning in areas such as computer vision, speech recognition, natural language processing, healthcare, robotics, and multimedia analysis.

The questions listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

1. Write a Python program to implement a Multi-Layer Perceptron (MLP) for classification using any standard dataset (e.g., Iris or MNIST).
   - Use at least one hidden layer.
   - Train the model and report accuracy and loss.
2. Write a Python program to implement and visualize the following activation functions:
   - Sigmoid
   - ReLU
   - Tanh

Plot their graphs and briefly compare their properties.
3. Implement a simple neural network using backpropagation to classify a dataset (e.g., Iris dataset).
Display:
   - Training loss
   - Accuracy

4. Design and train a Convolutional Neural Network (CNN) using TensorFlow/Keras for image classification on the MNIST dataset.
Tasks:
   - Define CNN layers
   - Train the model
   - Display test accuracy

5. Train a neural network model and demonstrate overfitting. Then apply one of the following techniques:
   - Dropout
   - Batch Normalization
   - Early stopping

Compare model performance before and after regularization.

6. Implement an RNN or LSTM model to predict the next value in a simple time-series dataset (e.g., stock prices or sine wave data).
7. Write a Python program to demonstrate a simple attention mechanism using matrix operations and visualize the attention weights.
8. Implement a simple Generative Adversarial Network (GAN) or Variational Autoencoder (VAE) to generate images using the MNIST dataset.
Display:

- Generated samples
- Training loss

9. Use a dataset of text (for example movie reviews) and build a text classification model using an RNN/LSTM.
Tasks:

- Preprocess text data
- Train the model
- Evaluate accuracy.

10. Use a pre-trained transformer model (such as BERT) using HuggingFace to perform one of the following tasks:

- Sentiment analysis
- Text summarization
- Question answering

Display the output for sample inputs.

Text/ Reference Books:

1. Deep Learning – Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press.
2. Pattern Recognition and Machine Learning – Christopher M. Bishop, Springer.
3. Neural Networks and Deep Learning – Michael Nielsen, Determination Press.
4. Deep Learning with Python – François Chollet, Manning Publications.
5. Speech and Language Processing – Daniel Jurafsky and James H. Martin, Pearson.
6. Dive into Deep Learning – Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Cambridge University Press.
7. Generative Deep Learning – David Foster, O'Reilly Media.
8. Natural Language Processing with Transformers – Lewis Tunstall, Leandro von Werra, and Thomas Wolf, O'Reilly Media.

Note: *Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*
a. *Implementation and execution - 8*
b. Viva- Voce – 5
c. Lab Assignment – 2

Paper: DRP-1 (PR): Dissertation

DRP-1: Dissertation : Total credit=12; This should be spilt into two parts: Dissertation preparation (08 credits) and presentation (04 credits);

In preparation section, a student will have to focus on a relevant topic *(1) Problem Definition (2) Literature Review, (3) Research Gap, (4) Objective(s), (5) Methodology, (6) Results, (7) Discussion (8) Conclusion and Future Scope and (9) References (following internationally accepted style)*

Guidelines: Each student of B.Sc. SEM - VIII will carry out one Dissertation under the supervision of a faculty member(s). The student will submit a report representing the actual work in a suitable format. The student should defend the report before the examiners. The work will be evaluated on the basis of presentation and viva-voce examination. It is expected that the research will be published in any reputed Conference / Book Chapter / Journal. The examination will be as per University guidelines.

Broad Areas: Real world problem solving using machine learning, Mobile app development, IOT based system, Natural Language processing etc. Following are some examples.

1. Predictive models, classification systems, etc. – Credit Risk Prediction using ML, Breast Cancer Early Detection, Customer Churn Prediction (with Explainable AI) etc.
2. Insights from large or complex datasets – Covid-19 Data Dashboard (Python + Plotly)
3. Projects involving sensors and microcontrollers – Smart Dustbin using Arduino + Ultrasonic Sensor, Home Automation with NodeMCU + Firebase, IoT-based Temperature Logger to Google Sheets
4. Text analytics, summarization, chatbots – Resume Ranker using NLP, Fake News Detection, Chatbot for College Enquiry

Note: *For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member(s) of the institution with seal. Distribution of marks will be as per the following.*

a. *(1) Problem Definition (Problem Definition, Literature Review, Research Gap): 40*
   (2) Implementation (Objective(S), Methodology, Results, Discussion, Conclusion and Future Scope): 80
   (3) Report (Citation, References and Report Writing): 30

b. *Project Presentation – 25*
c. *Viva- Voce – 50*

Signature of the BoS Members

Kaushik Roy
Dr. Kaushik Roy

Debadittya Barman
Dr. Debadittya Barman

Farhana Sultana 12/03/2026
Farhana Sultana

Debasmita Saha 12/03/26
Debasmita Saha

Arijit Bhattacharya
Arijit Bhattacharya

Akhil Kumar Das.
Dr. Akhil Kumar Das

Ekram Alam.
Ekram Alam