Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**B.Sc. (Computer Science)**
Class	**First Year**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-101-TH	**Subject-1**	Problem Solving using 'C' Programming	Theory	2	**02 Hours**

Course Objectives:

- *To introduce the foundations of computing, programming, and problem-solving using computers.*
- *To develop the ability to analyze a problem and devise an algorithm to solve it.*
- *To formulate algorithms, pseudocodes and flowcharts for arithmetic and logical problems.*
- *To understand structured programming approaches.*
- *To implement algorithms in the 'C' language.*
- *To test, debug and execute programs.*

Course Outcomes:

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

- *Explore algorithmic approaches to problem solving.*
- *Control the sequence of the program and give logical outputs.*
- *Understand and manage Input /Output operations in 'C' program*
- *Develop modular programs using control structures and arrays in 'C'.*

Unit	Title and Contents	No. of lectures in Clock Hours
1	**Chapter Name: Problem Solving Aspects** ● Introduction to problem solving using computers. ● Problem solving steps.	**Hours: 05**

	● Algorithms-definition, characteristics, examples, advantages and limitations. ● Flowcharts - definition, notations, examples , advantages and limitations, Comparison with algorithms. ● Programming Languages as tools, programming paradigms, types of languages ● Compilation process (compilers, interpreters), linking and loading, syntax and semantic errors, testing a program ● Good Programming Practices (naming conventions, documentation, indentation).	
2	**Chapter Name: 'C' Fundamentals** ● History of 'C' language. ● Application areas. ● Structure of a 'C' program. ● 'C' Program development life cycle. ● Function as building blocks. ● 'C' tokens ● Character set, Keywords , Identifiers ● Variables, Constants (character, integer, float, string, escape sequences, enumeration constant). ● Data Types (Built-in and user defined data types). ● Operators, Expressions, types of operators, Operator precedence and Order of evaluation. ● Character input and output. ● String input and output. ● Formatted input and output.	**Hours: 06**
3	**Chapter Name: Control Structures** ● Decision making structures:- if ,if-else, switch and conditional operator. ● Loop control structures:- while ,do while, for. ● Use of break and continue. ● Nested structures. ● Unconditional branching (goto statement).	**Hours: 06**
4	**Chapter Name: Arrays** ● Concept of array and also introduction of String. ● Types of Arrays – One, Two and Multidimensional array. ● Array Operations - declaration, initialization, accessing array elements. ● Memory representation of two-dimensional array (row major and column major) ● Array applications - Finding maximum and minimum, Counting occurrences, Linear search, Sorting an array elements, Matrix operations (trace of matrix, addition, multiplication, etc.)	**Hours: 06**
5	**Chapter Name: Basics of Function** ● Concept of function(Function Call, Definition and Declaration), Advantages of Modular design. ● Standard library functions.(math library functions, string library functions, etc.)	**Hours: 07**

	● Scope of variables and Storage classes.	

Reference Books:

1. How to Solve it by Computer, R.G. Dromey, Pearson Education.
2. Problem Solving and Programming Concept, Maureen Sprankle,7th Edition, Pearson Publication.
3. C: the Complete Reference, Schildt Herbert, 4th edition, McGraw Hill
4. A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard F. Gilberg, Cengage Learning India
5. The 'C' programming language, Brian Kernighan, Dennis Ritchie, PHI
6. Programming in C ,A Practical Approach, Ajay Mittal , Pearson
7. Programming with C, B. Gottfried, 3rd edition, Schaum's outline Series, Tata McGraw Hill.
8. Programming in ANSI C, E. Balagurusamy, 7th Edition, McGraw Hill.

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**B.Sc. (Computer Science)**
Class	**First Year**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-102-PR	**Subject-1**	**Lab Course based on CS-101-TH**	Practical	2	**04 Hours**

Course Objectives:

- *Explore and develop the algorithmic approaches to problem solving.*
- *Understand and implement modular programs using control structures and arrays in 'C'.*
- *Implement programming logic and also test, debug and execute programs.*
- *Implement Control the sequence of the program and give logical outputs.*

Course Outcomes:

On completion of this course, students will be able to:

- *Explore and develop the algorithmic approaches to problem solving.*
- *Understand and implement modular programs using control structures and arrays in 'C'.*
- *Implement programming logic and also test, debug and execute programs.*
- *Implement Control the sequence of the program and give logical outputs.*

Practical Guidelines:

Lab-Book: The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Submission:

Problem Solving Assignments:

The problem-solving assignments are to be submitted by the student in the form of a journal containing individual assignment sheets. Each assignment includes the Assignment Title, Problem statement, Date of submission, Assessment date, Assessment grade and instructors sign.

Programming Assignments:

Programs should be done individually by the student in the respective login. The codes should be uploaded on either the local server, Moodle, GitHub or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

Assessment:

Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes and good programming practices.

Operating Environment:

For 'C' Programming:
Operating system: Linux
Editor: Any linux based editor like vi, edit etc.
Compiler: cc or gcc

Assignment	Title and Contents
1	**Assignment Name: Problem Solving Aspects** ● Pseudo-code to programs. ● Compilation process (compilers , interpreters), linking and loading, syntax and semantic errors, testing a program ● Practices (naming conventions, documentation, indentation
2	**Assignment Name: 'C' Fundamentals** ● 'C' tokens and Character set, Keywords , Identifiers ● character, integer, float, string, escape sequences, enumeration constant. ● Built-in and user defined data types and Operators, Expressions, types of operators,Operator precedence and Order of evaluation.
3	**Assignment Name: Control Structures : Conditional Structures** ● Use of if ,if-else, ● Use of nested if, nested if-else.

	● Use of Switch case, nested Switch case ● Use of conditional operator
4	**Assignment Name: Control Structures : Loop Control Structures** ● Use of While loop ● Use of Do While loop ● Use of for loop ● Use of break and continue. ● Nested structures and goto statement.
5	**Assignment Name: One Dimensional Arrays** ● One Dimensional Arrays (1D) Operations - declaration, initialization, accessing array elements.
6	**Assignment Name: One Dimensional Arrays : Array Operations** ● Finding maximum and minimum, Counting occurrences, Linear search, etc.
7	**Assignment Name: One Dimensional Arrays : Sorting and Searching** ● Sorting an array (Simple exchange sort, (ie arrange the data in ascending and descending order))
8	**Assignment Name: Two Dimensional Arrays : Basic Operations** ● Two Dimensional Arrays (2D) Operations - declaration, initialization, accessing array elements.
9	**Assignment Name: Two Dimensional Arrays : matrix operations** ● Matrix operations : o Addition, o Subtraction o Multiplication, etc.
10	**Assignment Name: Functions** ● Use of Standard library functions:- function call, value returns. ● Use of math library functions, string library functions, etc.
11	**Assignment Name: Scope of variables** ● Use of Scope of variables ● Use of Storage classes.

Reference Books:

1. How to Solve it by Computer, R.G. Dromey, Pearson Education.
2. Problem Solving and Programming Concept, Maureen Sprankle,7th Edition, Pearson Publication.
3. C: the Complete Reference, Schildt Herbert, 4th edition, McGraw Hill
4. A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard F. Gilberg, Cengage Learning India
5. The ‘C’ programming language, Brian Kernighan, Dennis Ritchie, PHI
6. Programming in C ,A Practical Approach, Ajay Mittal , Pearson
7. Programming with C, B. Gottfried, 3rd edition, Schaum’s outline Series, Tata McGraw Hill.
8. Programming in ANSI C, E. Balagurusamy, 7th Edition, McGraw Hill.

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**B.Sc. (Computer Science)**
Class	**First Year**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CS-151-TH	**Subject-1**	Advanced 'C' Programming	Theory	2	2 Hours

Course Objectives:

- *To study advanced concepts of programming using the 'C' language.*
- *To understand code organization with complex data types and structures.*
- *To work with files.*

Course Outcomes:

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

- *Develop modular programs using control structures, function ,pointers, arrays, strings and structures*
- *Design and develop solutions to real world problems using C.*
- *Understand and repeat the sequence of instructions and points for a memory location.*
- *Identification, analyzation, development, verify and document the requirements for a computing environment.*

Unit	Title and Contents	No. of lectures in Clock Hours
1	**Chapter Name: Pointers** ● Introduction to Pointers. ● Declaration, definition, initialization, dereferencing. ● Pointer arithmetic. ● Relationship between Arrays & Pointers- Pointer to array, Array of pointers.	**Hours: 06**

	● Multiple indirection (pointer to pointer). ● Dynamic memory management- Allocation(malloc(),calloc()), Resizing(realloc()), Releasing(free())., ● Memory leak, dangling pointers.	
2	**Chapter Name : Functions** ● Concept of function, Advantages of Modular design. ● User defined functions:- declaration , definition, function call, parameter passing (by value , by reference), return statement. ● Recursive functions. ● Passing arrays to function. ● Functions and pointers- Passing pointer to function, Returning pointer from function, Function pointer.	**Hours: 05**
3	**Chapter Name: Strings** ● String Literals, string variables, declaration, definition, initialization. ● Array of strings. ● Strings and Pointers ● Command line arguments.	**Hours: 04**
4	**Chapter Name: Structures And Unions** ● Concept of structure, definition and initialization, use of typedef. ● Accessing structure members. ● Nested Structures ● Arrays of Structures ● Structures and functions- Passing each member of structure as a separate argument, Passing structure by value / address. ● Pointers and structures. ● Concept of Union, declaration, definition, accessing union members. ● Difference between structures and union.	**Hours: 08**
5	**Chapter Name: File Handling** ● Introduction to streams. ● Types of files. ● Operations on text files. ● Standard library input/output functions. ● Random access to files.	**Hours: 05**
6	**Chapter Name: Preprocessor** ● Role of Preprocessor ● Format of preprocessor directive ● File inclusion directives (#include) ● Macro substitution directive, argumented and nested macro ● Macros versus functions	**Hours: 02**
Reference Books:		

1. The Complete Reference, Schildt Herbert, 4th edition, McGraw Hill
2. A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard
3. Gilberg, Cengage Learning India
4. The ‘C’ programming language, Brian Kernighan, Dennis Ritchie, PHI R4. Programming in C ,A Practical Approach, Ajay Mittal , Pearson
5. Programming with C, B. Gottfried, 3rd edition, Schaum’s outline Series, Tata McGraw Hill.
6. Programming in ANSI C, E. Balagurusamy, 7th Edition, McGraw Hill.

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**B.Sc. (Computer Science)**
Class	**First Year**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CS-152-PR	**Subject-1**	**Lab Course based on CS-151-TH**	Practical	2	4 Hours

Course Objectives:

- *To study advanced concepts of programming using the 'C' language.*
- *To understand code organization with complex data types and programming structures.*
- *To work with files and its types.*

Course Outcomes:

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

- *Develop modular programs using control structures, function ,pointers, arrays, strings and structures*
- *Design and develop solutions to real world problems using Advanced C programming.*

Practical Guidelines:

LabBook: The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Submission:

Problem Solving Assignments:

The problem solving assignments are to be submitted by the student in the form of a journalcontainingindividualassignmentsheets.EachassignmentincludestheAssignmentTitle, Problem statement, Date of submission, Assessment date, Assessment grade and instructors sign.

Programming Assignments:
Programs should be done individually by the student in the respective login. The codes should be uploaded on either the local server, Moodle, Github or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

Assessment:

Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes and good programming practices.

Operating Environment:

For Advanced 'C' Programming:
Operating system: Linux
Editor: Any linux based editor like vi, edit etc. Compiler: cc or gcc

Assignment	Title and Contents
1	**Assignment Name: Pointers : Operations on pointers** ● Pointers - Declaration, ● definition, initialization, dereferencing ● Pointer arithmetic.
2	**Assignment Name: Pointers : Pointers and arrays]** ● Pointer to array, ● Array of pointers ● pointer to pointer
3	**Assignment Name: Pointers :Dynamic Memory allocation and dangling pointers and free** ● Dynamic memory management (Allocation) ● malloc(), calloc(), Resizing(realloc()) ● Releasing (free ())., ● dangling pointers.
4	**Assignment Name: Functions** User defined functions:- declaration , definition, function call, parameter passing (by value), return statement.
5	**Assignment Name: Passing Array to function (1D array and 2D array)** ● Passing 1D arrays to function ● Passing 2D arrays to function

6	**Assignment Name: Recursive Functions** • Use of Recursive functions
7	**Assignment Name: Pointers and Functions** • Passing pointer to function, • Returning pointer from function, • Function pointer
8	**Assignment Name: Strings : basic operations, array of strings & pointers and Command line arguments** • String Literals, string variables, declaration, definition, initialization and Syntax and use of user defined functions (e.g. – strlen, strcmp, strcpy, strcat) • Array of strings and Pointers • To access command-line arguments • Functions - atoi(), atol() and atof()
9	**Assignment Name: Structures : Basics** • Structure, definition and initialization, use of typedef. • Accessing structure members and Nested Structures
10	**Assignment Name: Arrays of Structures, Structure and Functions, Pointers and Structures** • Arrays of Structures and functions- Passing each member of structure as a separate argument, • Use of Pointers and Structures • Passing structure by value / address.
11	**Assignment Name: Unions** • Concept of Union, declaration, definition, accessing union members
12	**Assignment Name: File Handling** • Streams and Types of files. • Operations on text files. • Standard library input/output functions and Random access to files. • Accessing string and file using command line arguments
13	**Assignment Name: Preprocessor** • Preprocessor and Format of preprocessor directive • File inclusion directives (#include) • Macro substitution directive, argumented and nested macro and macros versus functions

Reference Books:

1. the Complete Reference, Schildt Herbert, 4th edition, McGraw Hill
2. A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard F. Gilberg, Cengage Learning India
3. The 'C' programming language, Brian Kernighan, Dennis Ritchie, PHI
4. Programming in C ,A Practical Approach, Ajay Mittal , Pearson

5. Programming with C, B. Gottfried, 3rd edition, Schaum's outline Series, Tata McGraw Hill.
6. Programming in ANSI C, E. Balagurusamy, 7th Edition, McGraw Hill.

***

The Poona Gujarati Kelavani Mandal's
HARIBHAI V. DESAI COLLEGE
of Arts, Science & Commerce (Autonomous)
Affiliated to Savitribai Phule Pune University
(Linguistic Minority Institution) AICTE NO. : 1-44457797714
ID No.: PU / PN / ASC / 057/ (1984)
NAAC Grade B++ (2.86 CGPA) ■ AISHE CODE : C-41829
Principal:
Dr. Rajendra G. Gurao
M.Sc., Ph.D.
Email: principal@hvdesaicollege.edu.in

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**B.C.A**
Class	**First Year**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-101-TH	Subject-1	**Problem Solving and Programming in C**	TH	2	2

- **Course Objectives:**
- *To provide a broad overview of problem-solving techniques.*
- *To learn C programming to solve problems*

Course Outcomes: *At the end of the course, students will be able to*

- *Define algorithms and explain their characteristics*
- *Formulate algorithm and draw flow chart to solve a given problem*
- *Explain use of appropriate data types, control statements*
- *Demonstrate ability to use top-down program design*

Unit	Title and content	No. of lectures in Clock Hours
Chapter-1	**Chapter Name: Problem solving, algorithms and flowcharts**	**Hours: 06**
• Types of Problems • Problem solving using computer • Difficulties with problem solving • Problem solving aspects. • Definition & Characteristics of algorithm • Examples of algorithms • Flow charts with examples • Top-down design		

- Problem solving using Arithmetic Statements, Conditional Statement & Iterative Statements such as Addition/Multiplication, check number is positive/negative, Maximum of 2 numbers & 3 numbers, sum of first n numbers, sum of given n numbers, reverse digits of a number, check whether the number is palindrome, check number is prime, factorial of number, factors of number etc.

Chapter-2	**Chapter Name: C Fundamentals**	**Hours: 07**

- Introduction to C, Features of C
- Structure of C Program
- C Character Set
- Identifiers and Keywords, Variables and constants
- Data types- Basic data types, Enumerated types
- Type casting, Declarations, Expressions
- Operators and Expressions Unary and Binary arithmetic operators, Increment Decrement operators, Relational and logical operators, Bit wise operators, Assignment operators, Comma operator, size of operator, Ternary conditional operator, Precedence and associativity
- Input Output Statements: printf, scanf functions, getchar, putchar, getch functions, gets, puts functions, Escape sequence characters, Format specifiers

Chapter-3	**Chapter Name: Control & Iterative Structures**	**Hours: 05**

- If, If- Else Statements
- Nested If Statements
- Conditional Branching – switch statement
- Loop statemets(while, do…while, for)
- break, continue, goto statements

Chapter-4	**Chapter Name: Functions**	**Hours: 06**

- Introduction to Functions
- Function Arguments
- Library & User defined functions,
- Methods for parameter passing
- Recursion
- Storage Classes – Auto, Static, Global and Register

Chapter-5	**Chapter Name: Arrays**	**Hours: 06**

- Introduction
- Array Declarations
- Bounds Checking
- Single dimension Arrays
- Two dimension Arrays
- Arrays & Function

Reference Books:

1. Cormen, Leiserson, Rivest, Stein, “Introduction to algorithms”
2. Brian W. Kernighan, Dennis M. Ritchie , “The C Programming Language”,ISBN:9788120305960, PHI Learning
3. R.G. Dromey, “How to Solve it by Computer”, ISBN: 9788131705629, Pearson Education
4. Behrouz A. Forouzan, Richard F. Gilberg, “A Structured Programming Approach Using C”, ISBN:9788131500941, Cengage Learning India
5. E. Balaguruswamy, “Programming in ANSI C”, ISBN: 9781259004612, Tata Mc-Graw Hill Publishing Co Ltd.-New Delhi
6. Maureen Spankle, “Problem Solving and Programming Concepts”, ISBN: 81-317-0711- 3
7. Y S Kanetkar, “Let Us C”, BPB Publications

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-102-PR	Subject-1	**Lab Course on CA-101 TH**	Practical	2	4

Course Objectives:

- *To learn formulation of algorithm for a given problem*
- *To study various data types, arrays and functions in C*
- *To understand input-output and, control and iterative statements in C*

Course Outcomes: *On completion of the course, students will be able to–*

- *Formulate an algorithm and draw flowchart for the given problem*
- *Implement the given algorithm in C*
- *Write programs using appropriate data types and control structures in C*

Unit	Title and content	No. of lectures in Clock Hours
Practical	Lab Course based on CA-101-TH	04
Assignment-1	Assignment on use of data types, simple operators (expressions)	
Assignment-2	Assignment on control statements I (if , if-else, switch case, break, continue)	
Assignment-3	Assignment on control statements II (Loops – for, while, do-while)	
Assignment-4	Assignment on nested loops	
Assignment-5	Assignment on writing C programs in modular way (use of user defined functions)	
Assignment-6	Assignment on call by value, call by reference and recursive functions.	
Assignment-7	Assignment on use of arrays (1-D array) and functions	
Assignment-8	Assignment on use of multidimensional array (2-D arrays) and functions	

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	**SEC-101-CA**	SEC (2)	**HTML and Web Page Designing**	PR	2	4

Course Objectives:

- *To understand web based application development process.*
- *To study basics of HTML elements and tag.*
- *To know usage of CSS in HTML.*
- *To design and create simple websites.*
- *To apply JavaScript to websites.*

Course Outcomes: *After successful completion of this course, learner will be able to*

- *Enlist various HTML elements and tags*
- *Use HTML elements and tags*
- *Apply CSS and Java script features.*
- *Design a website using HTML, CSS and JavaScript*

Unit	Title and content	No. of lectures in Clock Hours
Practical	HTML and Web Page Designing	04
Assignment-1	Using basic HTML elements (headings, paragraphs, line break, colour, fonts, links, Images, etc)	
Assignment-2	Creating Lists and Tables using HTML Tags	
Assignment-3	Creating Frames in HTML	
Assignment-4	Creating Forms using HTML	
Assignment-5	Designing of HTML screens using CSS	
Assignment-6	Using Functions in JavaScript	
Assignment-7	Carryout Validation and Event Handling using JavaScript	
Assignment-8	Designing website using basic elements of HTML, CSS and JavaScript.	

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	**OE-101-CA -TH**	GE/OE* (2)	**Introduction to Data Science**	TH	2	2

Course Objectives:

- *To understand need of Data Science*
- *To Know role of Statistics in Data Science*
- *To know Data Science Models and Tasks*

Course Outcomes: *At the end of the course, students will be able to*

- *Define Data Science Tasks and Models and Lifecycle*
- *Apply Prep-processing and visualization Techniques*

Unit	Title and content	No. of lectures in Clock Hours
	Introduction to Data Science	02
Chapter-1	**Chapter Name: Introduction**	**Hours: 06**
• What is Data Science? • Why learn Data Science? • Types of Data -structured, semi-structured, unstructured Data • Applications of Data Science • The Data Science Lifecycle • Role of Data Scientists • Data sources-Open Data, Social Media Data, Multimodal Data, standard datasets		
Chapter-2	**Chapter Name: Statistics for Data Science**	**Hours: 06**
• Data Objects and Attributes • Attribute Types: Nominal, Binary, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes, • Role of statistics in Data Science • Descriptive statistics - Measuring the Frequency • Measuring the Central Tendency: Mean, Median, and Mode, • Measuring the Dispersion: Range, Standard deviation, Variance, Inter quartile Range		

Chapter-3	Chapter Name: Data science Models and Tasks	Hours: 06
• Predictive and Descriptive Models • Introduction to Data Science Tasks – Classification, Prediction, Association, Clustering, • Performing simple Data Science Tasks using WEKA / R		
Chapter-4	**Chapter Name: Data Quality and Pre-processing**	**Hours: 06**
• Data Quality: Why Preprocess the Data?, Data munging/wrangling operations • Data Cleaning - Missing Values, Noisy Data • Data Transformation – Rescaling, Normalizing, • Data reduction and Data discretization		
Chapter-5	**Chapter Name: Data Visualization**	**Hours: 06**
• Introduction to Exploratory Data Analysis (EDA) • Data visualization, • Basic data visualization tools –Box Plots, Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts		

Reference Books:

1. Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020.
2. Data Mining Concepts and Techniques, Third Edition, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, 2012.
3. Data Mining Concepts and Techniques, Third Edition, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, 2012.

SEMESTER-II

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**B.C.A**
Class	**First Year**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-151-TH	Subject-1	**Advanced C Programming**	TH	2	2

Course Objectives:

- *To learn advanced features in C Programming*
- *To study advanced data types*
- *To understand built-in library functions*

Course Outcomes: *On completion of the course, student will be able to–*

- *Write programs using pointers and structures*
- *Use Pre-processor directives*
- *Manipulate strings using library functions*
- *Write programs to perform operations on Files*

Unit	Title and content	No. of lectures in Clock Hours
Chapter-1	**Chapter Name: Preprocessor**	**Hours: 06**
• Concept, Format of preprocessor directives, • File inclusion directives (#include), • Macro substitution directives (#define), • nested macros, parameterized macros, • Macros versus functions, • #error / #pragma directives, • Conditional compilation (#if/#ifdef/#else/#elif/#endif), • Predefined macros (_DATE_ / _TIME_ / _FILE_ / _LINE_ / _STDC_)		
Chapter-2	**Chapter Name: Pointers**	**Hours: 07**
• Concept – reference & dereference, • Declaration, definition, initialization & use, • Types of pointers,		

- Pointer Arithmetic,
- Multiple indirection,
- parameter passing – call by value and call by reference
- Arrays & Pointers - Pointer to array, Array of pointers,
- Functions & pointers - Passing pointer to function, Returning pointer from function,
- Pointers &const
- Dynamic memory management, Allocation, Resizing, Releasing, Memory leak / dangling pointersInput Output Statements: printf, scanf functions, getchar, putchar, getch functions, gets, puts functions, Escape sequence characters, Format specifiers

Chapter-3	**Chapter Name: S**	**Hours: 05**

- Concept, Declaration, definition, initialization,
- format specifiers,
- String literals/ constants & variables
- reading & writing from & to console,
- Importance of terminating NULL character,
- Strings & pointers
- Array of strings & array of character pointers,
- User defined functions,
- predefined functions in string.h - strlen , strcpy , strcat , strcmp , strcmpi , strrev , strlwr , strupr , strset , strchr , strrchr , strstr , strncpy , strncat , strncmp , strncmpi , strnset , strtok,
- Command line arguments – argc and argv

Chapter-4	**Chapter Name: Structures**	**Hours: 06**

- Concept, Declaration, definition, initialization,
- accessing structure members (. operator),
- Array of structures,
- Pointers to structures,
- Declaring pointer to structure
- Accessing structure members via pointer to structure,
- Structures & functions,
- Passing each member of structure as a separate argument,
- Passing structure by value / address,
- Nested structures & typedef structures,
- Concept of Union

Chapter-5	**Chapter Name: File Handling**	**Hours: 06**

- Concept of streams,
- need,
- Types of files,
- Operations on text & binary files,
- Random access file,
- library functions for file handling – fopen, fclose, fgetc, fseek, fgets, fputc etc

Reference Books:

8. The C Programming Language (Second Edition) – By B. W. Kerninghan& D. M. Ritchie
9. Programming in C – A Practical Approach – By Ajay Mittal (Pearson Publications)
10. Programming with C – By Byron S Gottfried (Schaum's Outlines)
11. A structural Programming Approach using C – By BehrouzForouzan& Richard Gilberg
12. Y S Kanetkar, "Let Us C", BPB PublicationsMaureen Spankle, "Problem Solving and Programming Concepts", ISBN: 81-317-0711- 3

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-152-PR	Subject-1	**Lab Course Based on CA-151-TH**	PR	4	4

Course Objectives:

- *To learn advanced features in C Programming*
- *To study advanced data types*
- *To understand built-in library functions*

Course Outcomes: *On completion of the course, students will be able to–*

- *Write programs using pointers and structures*
- *Use Pre-processor directives*
- *Manipulate strings using library functions*

Write programs to perform operations on Files

Unit	Title and content	No. of lectures in Clock Hours
Practical	**Lab Course based on CA-151-TH**	04
Assignment-1	To demonstrate use of preprocessor directives	
Assignment-2	To demonstrate use of pointers	
Assignment-3	To demonstrate advanced use of pointers	
Assignment-4	To demonstrate concept of strings, array of strings	
Assignment-5	To demonstrate string operations using pointers, command line arguments	
Assignment-6	To demonstrate structures (using array and functions)	
Assignment-7	To demonstrate nested structures	
Assignment-8	To demonstrate file handling	

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	SEC-151-CA-PR	SEC (2)	**Software Tools for Business Communications**	PR	2	4

Course Objectives:

- *To study word processing, spreadsheets and presentation tools*
- *To learn G-suit*
- *To be familiar with tools for Electronic communications*

Course Outcomes: *At the end of the course, students will be able to*

- *Perform various word processing tasks*
- *Prepare spreadsheets and presentations*
- *Collect feedbacks and make surveys*
- *Communicate and collaborate through electronic communications*

Unit	Title and Content	No. of lectures in Clock Hours
Chapter-1	**Chapter Name: Word processing and Google DOCs**	**No of Assignments: 02**
• Create, Save, Open and Edit Documents, • Text Alignments, Enhancements, and Effects • Basic Document Formatting and Editing, • Additional Document Formatting and Editing • Work with Multiple-Page Documents and Multiple Documents, • Work with Columns and Tables • Work with Objects, Lines, and Text Boxes, • Drawing Tools, • Add Special Effects • Create and manipulate Google DOC using various features		
Chapter-2	**Chapter Name: Spreadsheets and Google Sheets**	**No of Assignments: 02**
• Create, Save, and Print a Worksheet, • Use Formulas;		

- Copy a Formula;
- Format and Enhance
- Use Functions,
- Additional Formatting, and Editing,
- Create and Edit Charts,
- Integrate Worksheets with Other Applications
- Create and manipulate Google Sheets using various features

Chapter-3	Chapter Name: Presentations and Google Slides	No of Assignments: 01

- Create, Save, and Print a Presentation,
- Enhance Slides;
- Work with Text and Objects,
- Work with Slide Shows;
- Integrate Presentations with Other Applications
- Create and manipulate Google Slides using various features

Chapter-4	Chapter Name: Google Forms, Drives and Calendar	No of Assignments: 01

- Create, Save, Open and Edit Google form using essential features
- Google Drive: Create folders and subfolders, upload documents, share drive files and folders,
- Google Calendar: essential features

Chapter-5	Chapter Name: Emails, Groups and Generative AI Tools	No of Assignments: 02

- Create and send, receive emails,
- email folders and fields,
- attach documents, address book, email signatures and other essential settings,
- Email etiquettes
- Create, join email groups,
- send and receive emails on groups Using Generative AI tools such as ChatGPT

Reference Books:

1. Office 2019 in Easy Steps, Michael Price, BPB Publications
2. The Ridiculously Simple Guide to Google Apps (G Suite): A Practical Guide to Google Drive Google Docs, Google Sheets, Google Slides, and Google Forms, Scott La Counte, SL Editions)

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	OE-151-CA -PR	GE/OE* (2)	Data Science using Spreadsheets	PR	2	4

Course Objectives:

- *To know spreadsheet concepts*
- *To learn functions and formulas.*
- *To understand charts and graphics.*
- *To be familiar with filters and sorting of table data*

Course Outcomes : *After successful completion of this course, learner will be able to*

- *Perform computations on data using formulas.*
- *Present the data in graphical form.*
- *Analyze data by applying various functions and filters*

Unit	Title and content	No. of lectures in Clock Hours
Practical	Data Science using Spreadsheets	04
Assignment-1	To explore interface and basic features of Excel. Make a Start with Excel from simple to complex spreadsheet. Creating templates in Excel.	
Assignment-2	Using Autocomplete and formatting features. Data entry in Excel with different data types and formatting. Formatting Cells with Number formats, Font formats, Alignment, Borders, etc.	
Assignment-3	Printing Workbooks - Setting Up Print Area, Print Titles –Repeat Rows / Columns, Designing the structure of a template, Customizing Headers & Footers.	
Assignment-4	Filtering and Sorting - Filtering on Text, Numbers & Colours, Sorting Options, Sorting and Filtering Lists.	
Assignment-5	Calculations in MS-Excel using Basic Functions (Sum, Average, Max, Min, Count, etc). Use of Text Functions (Upper, Lower, Proper, Left, Mid,	

	Right , Trim, Len, Exact, Concatenate, Find, Substitute). Use of Arithmetic Functions (SumIf, SumIfs CountIf, CountIfs ,AverageIf, AverageIfs).
Assignment-6	What-If Analysis - Goal Seek, Data Tables, Solver Tool, Scenario Analysis.
Assignment-7	Data Validation- Number, Date & Time Validation, Dynamic Dropdown List Creation using Data Validation – Dependency List, Custom validations based on a formula for a cell, Text and List Validation.
Assignment-8	Generating different types of charts. Using SLICERS, Filter data with Slicers, Various Charts i.e. Bar Charts / Pie Charts / Line Charts, Manage Primary and Secondary Axis.
Assignment-9	Use of conditional functions. Applying IF functions. Conditional formatting in MS-Excel. Use of OFFSET function.
Assignment-10	Recording macros and buttons. Protecting Excel- Excel Security (File Level Protection Workbook, Worksheet Protection).
Assignment-11	Excel Dashboard, Planning a Dashboard, Adding Dynamic Contents to Dashboard, Adding Tables and Charts to Dashboard.
Assignment-12	Use of Lookup functions. (Vlookup / HLookup), Creating Smooth User Interface Using Lookup, Reverse Lookup using Choose Function
Assignment-13	Creating Simple Pivot Tables, Classic Pivot table, Basic and Advanced Value Field Setting, Calculated Field & Calculated Items, Grouping based on numbers and Dates.
Assignment-14	Arrays Functions - What are the Array Formulas, Use of the Array Formulas? Array with if, len, and mid functions formulas, Basic Examples of Arrays (Advanced Use of formulas with Array, Array with Lookup functions).

Reference Books:

4. Beginning Excel 2019, Authors: Noreen Brown, Barbara Lave, Julie Romey, Open Oregon Educational Resources
5. Excel Step by Step (Office 2021 and Microsoft 365) Published with the authorization of Microsoft Corporation by: Pearson Education, Inc.
6. Excel Bible: The Comprehensive Tutorial Resource
7. Excel: Quick Start Guide from Beginner to Expert (Excel, Microsoft Office)
8. Building Financial Models with Excel: A Guide for Business Professionals, (MISL-WILEY)
9. Predictive Analytics: Excel
10. Excel from Scratch: Excel course with demos and exercises

E-Resources:

1. https://www.udemy.com/course/microsoft-excel-2013-from-beginner-to-advanced-and-beyond/
2. https://edu.gcfglobal.org/en/excel/
3. https://support.microsoft.com/en-us/excel
4. https://www.coursera.org/projects/introduction-microsoft-excel
5. https://www.coursera.org/learn/microsoft-excel-work-smarter
6. https://www.udemy.com/course/excel-for-analysts/

***

The Poona Gujarati Kelavani Mandal's
HARIBHAI V. DESAI COLLEGE
of Arts, Science & Commerce (Autonomous)
Affiliated to Savitribai Phule Pune University
(Linguistic Minority Institution) AICTE NO. : 1-44457797714
ID No.: PU / PN / ASC / 057/ (1984)
NAAC Grade B++ (2.86 CGPA) ■ AISHE CODE : C-41829
Principal:
Dr. Rajendra G. Gurao
M.Sc., Ph.D.
Email: principal@hvdesaicollege.edu.in

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-501-MJ-TH	**Major Core**	Advanced Operating System	TH	4	4

Course Objectives:

- To learn Advanced Operating Systems Concepts
- To understand the programming interface to the Unix/Linux system
- To provide an understanding of the system calls of Operating Systems
- To get knowledge of the design and implementation of Operating Systems.

Course Contents:

Chapter-1	**Introduction to UNIX/Linux Kernel**	**Hours: 05**

1.1 System Structure
1.2 Architecture of UNIX Operating System
1.3 Introduction to System Concepts.
- Overview of file subsystem, processes, context of process, process states, state transitions, sleep and wakeup

Chapter-2	**Unix/Linux File Subsystem**	**Hours: 08**

2.1 Files and File System
2.2 Buffer Cache

- Buffer headers, Structure of the buffer pool, scenarios for retrieval of a buffer, reading and writing disk blocks, advantages and disadvantages of buffer cache.

2.3. Internal Representation of Files

- Inodes, Structure of regular file, Directories

Chapter-3	**System Calls for File Subsystem**	**Hours: 12**

3.1 File I/O System calls

- open, read, write, lseek, close, creat, pipes, dup

3.2 File Access System calls

- Atomic operations, dup2, sync, fsync, and fdatasync, fcntl, /dev/fd
- stat, fstat, lstat, file types, Set-User-ID and Set-Group-ID, file access permissions, ownership of new files and directories, access function, umask function, chmod and fchmod, sticky bit, chown, fchown, and lchown, file size, file truncation, file systems, link, unlink, remove, and rename functions, symbolic links, symlink and readlink functions, file times, utime, mkdir and rmdir, reading directories, chdir, fchdir, and getcwd, device special files

Chapter-4	**Unix/Linux Process Control Subsystem**	**Hours: 12**

4.1 Process states and transitions

4.2 Layout of system memory

- Regions, Pages and Page tables, Layout of Kernel, Uarea

4.3 Context of a process

4.4 Saving the context of a process

- Interrupts and Exceptions, System Call Interface, Context Switch

4.5 Sleep

- Sleep events and addresses, Algorithms for Sleep and Wakeup

4.6 Process creation

4.7 Process termination

4.8 Awaiting process termination

4.9 Invoking other programs

4.10 The user id of a process

4.11 Changing the size of the process

4.12 System Book and Init Process

Chapter-5	**System Calls Process Control Subsystem**	**Hours: 08**

5.1 Process Environment System Calls

- setjmp and longjmp, getrlimit and setrlimit

5.2 Process Control System Calls

- fork, vfork, exit, wait and waitpid, waitid, wait3 and wait, exec, changing user IDs and group IDs, system function, user identification, process times
- Process groups

Chapter-6	Signal Handling	Hours: 07
6.1 Introduction 6.2 Signal Concepts 6.3 Signal function 6.4 kill and raise functions 6.5 alarm and pause functions 6.6 abort function 6.7 sleep function		
Chapter-7	**Memory Management**	**Hours: 08**
7.1 Swapping - Allocation of swap space, Swapping process out, Swapping process in 7.2 Demand Paging - Data structures for demand paging, Page stealer process, Page faults		
Reference Books: 1. Maurice J. Bach.; The Design of the UNIX Operating System; PHI 2. Richard Stevens; Advanced Programming in the UNIX Environment; Addison-Wesley 3. Robert Love; Linux System Programming; O"Reilly		
Examination Scheme	**CIE : 30 Marks** **SEE : 70 Marks**	

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-502-MJ-TH	**Major Core**	Artificial Intelligence	TH	4	4

Course Objectives:

- To understand the concept of Artificial Intelligence (AI) in the form of various tasks.
- To understand Problem Solving using various searching strategies for AI.
- To understand multi-agent environment.
- To acquaint with the fundamentals of knowledge and reasoning.
- To understand Fundamentals of Game Theory.
- To explore of AI applications.

Course Outcomes:

On Completion of this course, student will be able to -

CO1: Understand the fundamental concepts of Artificial Intelligence.

CO2: Identify and apply appropriate search strategies for AI problem.

CO3: Identify knowledge and represent AI algorithms using various techniques.

CO4: Implement ideas to design and develop AI solutions for complex challenges.

CO5: Analyze the performance of AI models and interpret their results.

CO6: Implement ideas underlying modern logical inference systems.

CO7: Understand recent trends and future scope of AI.

Course Contents:

Chapter-1 I	**Introduction to Artificial Intelligence**	**Hours: 06**

1.1 Introduction to Artificial Intelligence
1.2 Foundations of Artificial Intelligence
1.3 History of Artificial Intelligence
1.4 AI Risks and Benefits
1.5 Characteristics of Intelligent Agents
1.6 Structure of Agents
1.7 Agents and Environments
1.8 Types of Intelligent Agents.

Chapter-2	**Problem Solving**	**Hours: 10**

2.1 Problems Solving methods
2.2 Problem-Solving Agents
2.3 Example Problems
2.4 Search Algorithms
2.5 Blind Search Techniques: -BFS, DFS, DLS, Iterative Deepening, Search, Bidirectional Search, Uniform cost Search.
2.6 Heuristic search techniques: -Generate and test,Hill Climbing, Best First search, Constraint Satisfaction, Mean-End Analysis, A*,AO*.

Chapter-3	**Game Theory**	**Hours: 10**

3.1 Optimal Decisions in Games
3.2 Heuristic Alpha–Beta Tree Search
3.3 Monte Carlo Tree Search
3.4 Stochastic Games
3.5 Partially Observable Games
3.6 Limitations of Game Search Algorithms
3.7 Constraint Satisfaction Problems (CSP).

Chapter-4	**Knowledge Representation**	**Hours: 10**

4.1 Representations and Mappings
4.2 Approaches to Knowledge Representation
4.3 Knowledge representation method
4.4 Logical Agents
4.5 Knowledge-Based Agents
4.6 Logic, Propositional Logic
4.7 Effective Propositional Model Checking

4.8 Predicate logic
4.9 Representing Simple facts in Logic.

Chapter-5	**Reasoning**	**Hours: 10**

5.1 Inference in First-Order Logic
5.2 Propositional vs. First-Order Inference
5.3 Unification and First-Order Inference
5.4 Forward Chaining, Backward Chaining
5.5 Resolution
5.6 Categories and Objects
5.7 Events
5.8 Mental Objects and Modal Logic
5.9 Reasoning Systems for Categories
5.10 Reasoning with Default Information

Chapter-6	**Planning**	**Hours: 08**

6.1 Classical Planning
6.2 Automated Planning
6.3 Algorithms for Classical Planning
6.4 Heuristics for Planning
6.5 Hierarchical Planning
6.6 Planning and Acting in Nondeterministic Domains Time, Schedules, and Resources
6.7 Analysis of Planning Approaches

Chapter-7	**Recent trends in AI**	**Hours: 06**

7.1 Applications of AI
7.2 Language model
7.3 Information retrieval
7.4 Information Extraction
7.5 Introduction to Natural Language Processing (NLP)
7.6 Reinforcement Learning and Robotics
7.7 Computer Vision Breakthroughs
7.8 AI in Healthcare
7.9 AI in Finance Autonomous Systems.
7.10 Introduction to Explainable AI
7.11 Introduction to Generative AI

Reference Books:

1. S. Russell and P. Norvig,"Artificial Intelligence: A Modern approach" , Prentice Hall , Third

<table>
<tr><td colspan="2">edition,2009.
2. Computational Intelligence Eberhart Elsevier Publication
3. Artificial Intelligence: A New Synthesis Nilsson Elsevier Publication
4. Artificial Intelligence with Python PrateekJoshi Packt Publishing Ltd
5. Artificial Intelligence Saroj Kausik Cengage Learning
6. Nilsson Nils J , “Artificial Intelligence: A new Synthesis”, Morgan Kaufmann Publishers Inc. San Francisco, CA, ISBN: 978-1-55-860467-4
7. Patrick Henry Winston, “Artificial Intelligence”, Addison-Wesley Publishing Company, ISBN: 0-201-53377-4 .
8. Andries P. Engelbrecht-Computational Intelligence: An Introduction, 2nd Edition-Wiley India- ISBN: 978-0-470-51250-0</td></tr>
<tr><td>Examination Scheme</td><td>CIE : 30 Marks
SEE : 70 Marks</td></tr>
</table>

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	**CS-503-MJ-TH**	**Major Core**	**Principles of Programming Languages**	**TH**	*2*	*2*

Course Objectives:

- To introduce the various programming paradigms.
- To understand the evolution of programming languages.
- To understand the concepts of OO languages, functional languages, logical and scripting languages

Course Outcomes:

On Completion of this course, student will be able to – think about programming languages analytically:

CO1: Separate syntax from semantics
CO2: Compare programming language designs
CO3: Understand their strengths and weaknesses
CO4: Learn new languages more quickly
CO5: Understand basic language implementation techniques
CO6: Learn small programs in different programming Languages

Course Contents:

Chapter-1	**Introduction**	**Hours: 02**

1.1 The Art of Language Design
1.2 The Programming Language Spectrum
1.3 Why Study Programming Languages?

1.4 Compilation and Interpretation
1.5 Programming Environments

Chapter-2	**Names, Scopes, Bindings, Object Orientation Concepts**	**Hours: 06**

2.1 The Notion of Binding Time.
2.2 Object Lifetime and Storage Management.
2.3 Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection , Scope Rules
2.4 Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping, The meaning of Names in a Scope
2.5 Object-Oriented Programming
2.6 Encapsulation and Inheritance, Modules, Classes, Nesting (Inner Classes), Type Extensions, Extending without Inheritance
2.7 Initialization and Finalization, Choosing a Constructor, References and Values, Execution Order, Garbage Collection
2.8 Dynamic Method Binding
2.9 Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup, Polymorphism, Object Closures
2.10 Multiple Inheritance, Shared Inheritance, Mix-In Inheritance
2.11 Semantic Ambiguities, Replicated Inheritance

Chapter-3	**Data Types**	**Hours: 08**

3.1 Introduction
3.2 Primitive Data Types
3.3 Numeric Types: Integer, Floating point, Complex, Decimal, Boolean Types, Character Types.
3.4 Character String Types
3.5 Design Issues, Strings and Their Operations, String Length Operations, Evaluation, Implementation of Character String Types
3.6 User defined Ordinal types, Enumeration types, Design Evaluation Subrange types, Ada‟s design Evaluation Implementation of user defined ordinal types.
3.7 Array types.
3.8 Design issues, Arrays and indices, Subscript bindings and array categories, Heterogeneous arrays, Array initialization, Array operations, Rectangular and Jagged arrays, Slices, Evaluation, Implementation of Array Types
3.9 Associative Arrays: Structure and operations, Implementing associative arrays.
3.10 Record types: Definitions of records, References to record fields, Operations on records, Evaluation, implementation of Record types
3.11 Union Types: Design issues, Discriminated versus Free unions, Evaluation,

Implementation of Union types.

3.12 Pointer and Reference Types :Design issues, Pointer operations, Pointer problems, Dangling pointers, Lost heap dynamic variables, Pointers in C and C++, Reference types, Evaluation

3.13 Implementation of pointer and reference types

3.14 Representation of pointers and references, Solution to dangling pointer problem, Heap management

Chapter-4	**Control Flow**	**Hours: 06**

4.1 Expression Evaluation, Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation.

4.2 Structured and Unstructured Flow, Structured Alternatives to goto Sequencing.

4.3 Selection - Short-Circuited Conditions, Case/Switch Statements, Iteration.

4.4 Iteration - Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops Recursion

4.5 Recursion - Iteration and Recursion, Applicative- and Normal-Order Evaluation

Chapter-5	**Subprograms and Implementing Subprograms**	**Hours: 08**

5.1 Introduction

5.2 Fundamentals of Subprograms

5.3 Design Issues for subprograms

5.4 Local Referencing Environments

5.5 Parameter-Passing Methods

5.6 Parameters That Are Subprograms

5.7 Overloaded Subprograms

5.8 Generic Subroutines, Generic Functions in C++, Generic Methods in Java

5.9 Design Issues for Functions

5.10 User-Defined Overloaded Operators Coroutines

5.11 Implementing Subprograms

5.12 The General Semantics of Calls and Returns

5.13 Implementing "Simple" Subprograms

5.14 Implementing Subprograms with Stack- Dynamic Local Variables

5.15 Nested Subprograms Blocks

5.16 Implementing Dynamic Scoping

Reference Books:

1. Michel L. Scott; Programming LanguagePragmatics, 3e; Kaufmann Publishers, An Imprint of Elsevier, USA
2. Robert W. Sebesta; Concepts of ProgrammingLanguages, Eighth Edition; Pearson Education
3. Alvin Alexander; Scala Cookbook; O"REILLY publication

Examination Scheme	**CIE : 15 Marks** **SEE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-504-MJ-PR	**Major Core**	Lab course on CS-501-MJ-TH	PR	2	4

Course Objectives:

- To learn Advanced Operating Systems Concepts
- To understand the programming interface to the Unix/Linux system
- To provide an understanding of the functions of Operating Systems
- To get knowledge of the design and implementation of Operating Systems.

Course Outcomes:

On Completion of this course, student will be able to -

CO1: Understand the Operating Systems Structure with example of Unix/Linux.

CO2: Learn the structure of files and directory in UNIX/LINUX OS.

CO3: Use various system calls related to file subsystem.

CO4: Learn the process control subsystem structure in UNIX/LINUX OS

CO5: Use various system calls related to process control subsystem.

CO6: Learn the concept of signal handling with practical implementation

Course Contents:

Assign No.	Practical Assignment using C Programming

1.	Create a file with hole in it.
2.	Take multiple files as Command Line Arguments and print their inode number
3.	Write a C program to find file properties such as inode number, number of hard link, File permissions, File size, File access and modification time and so on of a given file using stat() system call.
4.	Print the type of file where file name accepted through Command Line
5.	Write a C program to find whether a given file is present in current directory or not.
6.	Write a C program that a string as an argument and return all the files that begins with that name in the current directory. For example > ./a.out foo will return all file names that begins with foo
7.	Read the current directory and display the name of the files, no of files in current directory
8.	Write a C program which receives file names as command line arguments and display those filenames in ascending order according to their sizes. I) (e.g $ a.out a.txt b.txt c.txt, …)
9.	Display all the files from current directory which are created in particular month
10.	Display all the files from current directory whose size is greater that n Bytes Where n is accept from user.
11.	Write a C Program that demonstrates redirection of standard output to a file.
12.	Write a C program that will only list all subdirectories in alphabetical order from current directory.
13.	Write a C program that redirects standard output to a file output.txt. (use of dup and open system call).
14.	Write a C program to Identify the type (Directory, character device, Block device, Regular file, FIFO or pipe, symbolic link or socket) of given file using stat() system call.
15.	Generate parent process to write unnamed pipe and will read from it
16.	Handle the two-way communication between parent and child processes using pipe.
17.	Demonstrate the use of atexit() function.
18.	Write a C program to demonstrates the different behaviour that can be seen with automatic, global, register, static and volatile variables (Use setjmp() and longjmp() system call).
19.	Implement the following unix/linux command (use fork, pipe and exec system call) ls –l | wc –l
20.	Write a C program to create „n“ child processes. When all „n“ child processes terminates, Display total cumulative time children spent in user and kernel mode.
21.	Write a C program to create an unnamed pipe. The child process will write following three messages to pipe and parent process display it. Message1 = “Hello World” Message2 = “Hello SPPU” Message3 = “Linux is Funny”
22.	Write a C program to get and set the resource limits such as files, memory associated with a process

23.	Write a program that illustrates how to execute two commands concurrently with a pipe.
24.	Write a C program that print the exit status of a terminated child process
25.	Write a C program that catches the ctrl-c (SIGINT) signal for the first time and display the appropriate message and exits on pressing ctrl-c again.
26.	Write a C program which creates a child process and child process catches a signal SIGHUP, SIGINT and SIGQUIT. The Parent process send a SIGHUP or SIGINT signal after every 3 seconds, at the end of 15 second parent send SIGQUIT signal to child and child terminates by displaying message "My Papa has Killed me!!!".
27.	Write a C program to send SIGALRM signal by child process to parent process and parent process make a provision to catch the signal and display alarm is fired.(Use Kill, fork, signal and sleep system call)
28.	Write a C program that illustrates suspending and resuming processes using signals.
29.	Write a C program which create a child process which catch a signal sighup, sigint and sigquit. The Parent process send a sighup or sigint signal after every 3 seconds, at the end of 30 second parent send sigquit signal to child and child terminates my displaying message "My DADDY has Killed me!!!".
30.	Write a C program to implement the following unix/linux command (use fork, pipe and exec system call). Your program should block the signal Ctrl-C and Ctrl-\ signal during the execution. i. Ls –l | wc –l
31.	Write a C program which creates a child process to run linux/ unix command or any user defined program. The parent process set the signal handler for death of child signal and Alarm signal. If a child process does not complete its execution in 5 second then parent process kills child process.

Examination Schem-CIE : 15 Marks

SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-505-MJ-PR	**Major Core**	Lab course on CS-502-MJ-TH	PR	2	4

Course Objectives:

- To understand the concept of Artificial Intelligence (AI) in the form of various tasks.
- To understand Problem Solving using various searching strategies for AI.
- To understand multi-agent environment.
- To acquaint with the fundamentals of knowledge and reasoning.
- To understand Fundamentals of Game Theory.
- To explore of AI applications.

Course Outcomes:

On Completion of this course, student will be able to -

CO1: Understand the fundamental concepts of Artificial Intelligence.

CO2: Identify and apply appropriate search strategies for AI problem.

CO3: Identify knowledge and represent AI algorithms using various techniques.

CO4: Implement ideas to design and develop AI solutions for complex challenges.

CO5: Analyze the performance of AI models and interpret their results.

CO6: Implement ideas underlying modern logical inference systems.

CO7: Understand recent trends and future scope of AI.

Course Contents:

Sr n	Practical Assignment

o	
1.	Practical on basic programs using python for introducing and using python environment such as, a) Program to print multiplication table for given no. b) Program to check whether the given no is prime or not. c) Program to find factorial of the given no and similar programs.
2.	Write a program to implement List Operations Nested list, Length, Concatenation, Membership ,Iteration ,Indexing and Slicing List Methods Add, Extend & Delete
3.	Write a program to Illustrate Different Set Operations.
4.	Write a program to implement Simple Chatbot.
5.	Write a program to implement Breadth First Search Traversal
6.	Write a program to implement Depth First Search Traversal.
7.	Write a program to implement Water Jug Problem
8.	Write a program to implement K -Nearest Neighbor algorithm.
9.	Write a program to implement Regression algorithm
10.	Write a program to implement Random Forest Algorithm
11.	Develop a program to solve the eight queens problem. (Uninformed Search)
12.	Implement a system that performs arrangement of some set of objects in a room. Assume that you have only 5 rectangular, 4 square-shaped objects. Use A* approach for the placement of the objects in room for efficient space utilisation. Assume suitable heuristic, and dimensions of objects and rooms. (Informed Search)
13.	Implement a program for learning agent for a lift, where The lift would halt at a particular floor based on the identity of the individual. There would be energy optimisation through elimination of redundant operation. (Intelligent Agent)
14.	Develop a program to solve the N queens puzzle using forward checking. Show in steps how the constraints are handled. (Constraint Satisfaction Problem)
15.	Write a computer program to play tic-tac-toe game. (Game Theory)

Examination Scheme	**CIE : 15 Marks** **SEE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	**CS-510-MJ-TH**	**Major Elective**	**Advance Databases and Web Technologies**	**TH**	**2**	**2**

Course Objectives

- Provides an overview of the concept of NoSQL technology.
- Provides an insight into the different types of NoSQL databases
- Makes the student capable of making a choice of what database technologies to use, based on their application needs.
- To introduce students to modern web technologies.
- To introduce students to modern web designing technologies.
- Should gain knowledge about web designing using html5 and css3

Course Outcomes

On Completion of this course, student will be able to -

CO1: Students will get knowledge of advance database technology

CO2: Students will be able to choose appropriate database technology as per application

CO3: Students will learn to design responsive web application

CO4: Students could design and implement scalable web application

Course Contents:

Chapter-1	Introduction to NOSQL	Hours: 05

1.1 Database Concept
1.2 Relational Databases
1.3 Introduction to the NoSQL database
1.4 Why NoSQL
1.5 Features of NOSQL
1.6 Aggregate Data Models
1.7 Distribution Models
1.8 Approaches to data distribution

Chapter-2	NOSQL Databases	Hours: 09

2.1 Schema Migration
2.2 Polyglot Persistence
2.3 Introduction to Key-Value Databases (Riak) Concept,
Features, Use Cases
2.4 Introduction to Column Family
Stores (Cassandra) Concept, Features, Use Cases
2.5 MongoDB The Document Data Model, Documents and Collections, MongoDB Use Cases, Embedded Data Models, Replication via Replica Sets, MongoDB Design, MongoDB and the CAP Theorem, The MongoDB Data Manipulation Language, Transactions, Atomicity, and Documents
2.6 Introduction to Graph databases (Neo4j)
Overview of Graph Theory, The Graph Data Model, Graph Database Use Cases, Neo4j Design: Standalone and Cluster, ACID Properties and the CAP Theorem, CRUD Operations with the Neo4j Core API, Navigating Graphs with the Traversal API, The Neo4j REST API, The Cypher Data Manipulation Language, Querying as Graph Traversal

Chapter-3	Basics of HTML5	Hours: 04

3.1 Introduction
3.2 Semantic Elements
<article>, <aside>, <figcaption>, <figure>, <footer>, <header>, <mark>, <nav>
<progress>, <section>, <summary>, <time>
3.3 Form Elements
<datalist>, <keygen>, <output>
3.4 Form Input Types
Color, Date, Datetime, Datetime-local, Email, Month, Number, Range, Search, Tel, Url, Time, Week
3.5 Form Attributes

Autocomplete, autofocus, form, formaction, formenctype, formmethod, formnovalidate, Formtarget

Chapter-4	**CSS3 Introduction**	**Hours: 04**

4.1 Introduction

Borders, border-radius, Border Images, Backgrounds, Background Size, background- origin, Text Effects, text-shadow, box-shadow, Text, text-overflow, word-wrap, word- break, Fonts

4.2 Transformations

2D Transforms, 3D Transforms

4.3 Transitions

transition-delay, transition-duration, transition-property, transition-timing-function

Chapter-5	**Introduction to BootStrap**	**Hours: 08**

5.1 Overview of Bootstrap

Introduction of Bootstrap, Syntax of Bootstrap, Container and Container-fluid, Connectivity of Bootstrap in page

5.2 Bootstrap Component

Jumbotron, Button, Grid, Table, Form, Alert, Wells, Badge and label, Panels, Pagination, Pager, Image, Glyph icon, Carousel, Progress Bar, List Group, Dropdown, Collapse

5.3 Bootstrap Advance Component

Tabs/Pill, Navbar, Input Types, Modals, Popover, Scrollspy,

5.4 Bootstrap Utilities

Bootstrap Border, Bootstrap Clearfix, Bootstrap Close Icons, Bootstrap Colors, Display Flexbox, Display Property, Image Replacement, Invisible Content, Bootstrap Position, Responsive helpers,Screen Readers, Bootstrap sizing, Bootstrap spacing, Bootstrap Typography.

Reference Books:

1. Sadalage, P. & Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. (1st Ed.). Upper Saddle River, NJ: Pearson Education, Inc. ISBN- 13: 978-0321826626 ISBN-10: 0321826620
2. Redmond, E. & Wilson, J. (2012). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement (1st Ed.). Raleigh, NC: The Pragmatic Programmers, LLC. ISBN-13: 978-1934356920 ISBN-10: 1934356921
3. Dan Sullivan, "NoSQL For Mere Mortals", 1st Edition, Pearson Education India, 2015. (ISBN13: 978-9332557338)
4. Head First HTML5 Programming: Building Web Apps with JavaScript Book by Elisabeth Robson and Eric Freeman
5. HTML5 and CSS3: Building Responsive Websites Book by Ben Frain and Benjamin LaGrone

Examination Scheme	**CIE : 15 Marks**

	SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-511-MJ-PR	*Major Elective*	Lab Course on CS-510-MJ-TH	PR	2	4

Objectives

- Provides an overview of the concept of NoSQL technology.
- Provides an insight into the different types of NoSQL databases
- Makes the student capable of making a choice of what database technologies to use, based on their application needs.
- To introduce students to modern web technologies.
- To introduce students to modern web designing technologies.
- Should gain knowledge about web designing using html5 and css3

Student able to use frame work

Course Outcomes

On Completion of this course, student will be able to -

CO1: Students will get knowledge of advance database technology

CO2: Students will be able to choose appropriate database technology as per application

CO3: Students will learn to design responsive web application

CO4: Students could design and implement scalable web application

Course Contents:

Assign No.	Practical Assignment
1-10	**MongoDB Practical Assignment** 1. Create a Employee collection with mentioned fields

	Employee (eno,ename,salary,desig,dept:{deptno,deptname,location}, project:{pname,hrs}) 2. Insert 10 documents in Employee collection 3. Display all the documents from Employee collection 4. Display all employees whose name starts with „S" 5. Display all Employee with the designation „Manager" 6. Display all employees with salary >50000 and salary <80000 7. Update no. of hrs to 7 for pname=__ 8. Add bonus Rs. 5000 for all employees with salary >50000 and salary <150000 9. Increase salary by 20% of employees working in deptname=_____ 10. Remove all employees working on pname=_
11-13	**Neo4j Practical Assignment** 11. Library Database : i. List all people, who have issued a book "……" ii. Count the number of people who have read " …." iii. Add a property "Number of books issued " for Mr. Joshi and set its value as the count iv. List the names of publishers from pune city. 12. Song Database: i. List the names of songs written by ":….." ii. List the names of record companies who have financed for the song "…." iii. List the names of artist performing the song "….." iv. Name the songs recorded by the studio " ……." 13. Library database a) List all readers who have recommended either book "…" or " " or "" b) List the readers who haven"t recommended any book c) List the authors who have written a book that has been read / issued by maximum numberof readers. d) List the names of books recommended by " " And read by at least one reader e) List the names of books recommended by " " and read by maximum number of readers. f) List the names of publishers who haven"t published any books written by authors fromPune and Mumbai. g) List the names of voracious readers in our library **.....**
14-18	**Web Technology Assignment** 14. Create an HTML5 program for the following input type i. Date time ii. email input type iii. search input type

	15. Write an 5 program for student registration for college admission. 16. Write a css3 script for the above student registration form e.g. high lite compulsory fields in a different color 17. Write a bootstrap program for the following "The table class adds basic styling (light padding and only horizontal dividers) to a table" The table can have the first name, last name, and email id as columns. 18. Write a bootstrap application to display thumbnails of the images
Examination Scheme	**CIE : 15 Marks** **SEE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-512-MJ-TH	Major Elective	Cloud Computing	TH	2	2

Objectives

- To understand the principles and paradigm of Cloud Computing
- To appreciate the role of Virtualization Technologies
- Ability to design and deploy Cloud Infrastructure

Understand cloud security issues and solutions

Course Outcomes

On Completion of this course, student will be able to –

CO1: To understand the principles of cloud computing

CO2: To understand the importance of virtualization and how it has helped the development of cloud computing.

CO3: To understand the concept of cloud security.

CO4: To design and deploy cloud infrastructure.

CO5: To understand the concept of edge computing

Course Contents:

Chapter-1	Introduction to Cloud Computing	Hours: 08

1.1 Overview & Evolution
Computing
Types of computing

Distributed Computing, Grid Computing, Cluster Computing, Utility Computing Introduction to Cloud Computing

Features/Characteristics of a cloud

Advantages & Disadvantages of Cloud Computing. Challenges of cloud computing

1.2 Cloud Architecture

Deployment Models

Public, Private, Hybrid and Community Cloud Service Models

Infrastructure as a Service, Platform as a Service, Software as a Service, Everything as a Service.

1.3 Cloud Service providers

1.4 Cloud Enabling Technologies

Broadband networks and internet architecture

Data centre technology

Virtualization technology

Web technology

Multitenant technology

Chapter-2	**Abstraction and Virtualization**	**Hours: 05**

2.1 Virtualization Technologies

Introduction to virtualization, Types of Virtualization Benefits and Disadvantages of Virtualization

2.2 Load Balancing & Virtualization What is

Load Balancing Working of Load Balancers

Advantages of Load Balancing

2.3 Hypervisors & its types

2.4 Virtual Machines Provisioning and Migration Services Virtual Machine Provisioning

Virtual Machine Life Cycle/ VM Provisioning Process Virtual Machine Migration Services

VM Migration and need

VM Migration Techniques/Methods Cloud Provisioning

Types of Cloud Provisioning Virtualization of CPU, Memory & I/O Devices

2.5 Virtual Clusters and Resource Management

2.6 Physical v/s Virtual Clusters

2.7 Resource Management

Chapter-3	**Overview of Cloud Security**	**Hours: 08**

3.1 Overview of Cloud Security Cloud Security Threads

Cloud Security Challenges and Risks

3.2 Security Architecture Design
Infrastructure Security
Data Security Application Security
Virtual Machine Security
3.3 Cloud Security Monitoring Security
Monitoring Benefits & Challenges
3.4 Identity Management and Access Control
Identity Management
Multi-Factor Authentication(MFA)
Identity Verification
Authentication, Authorization, and Accountability (AAA)
3.5 Disaster Recovery in Clouds.

Chapter-4	**Cloud Technologies and Advancements**	**Hours: 09**

4.1 Features of Cloud and Grid platforms
4.2 Programming support for Google App Engine
4.3 Programming on Amazon AWS
4.4 Programming on Microsoft Azure
4.5 Emerging Cloud software Environments
4.6 Understand the need of Cloud Computing
4.7 Existing Cloud Applications and opportunities for new Applications

Reference Books:

1. Brian J.S. Chee and Curtis Franklin : Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center
2. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi : Mastering Cloud Computing: Foundations and Applications Programming
3. Kai Hwang, Geoffrey C Fox, Jack G Dongarra : Distributed and Cloud Computing, From Parallel Processing to the Internet of Things

Examination Scheme	**CIE : 15 Marks** **SEE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-513-MJ-PR	Major Elective	Lab Course on CS-512-MJ-TH	PR	2	4

Objectives

1. To understand the principles and paradigm of Cloud Computing
2. To appreciate the role of Virtualization Technologies
3. Ability to design and deploy Cloud Infrastructure
4. Understand cloud security issues and solutions

Course Outcomes

On Completion of this course, student will be able to –

CO1: To understand the principles of cloud computing

CO2: To understand the importance of virtualization and how it has helped the development of cloud computing.

CO3: To understand the concept of cloud security.

CO4: To design and deploy cloud infrastructure.

Course Contents:

Assign No.	Practical Assignment
1.	Working and Implementation of Infrastructure as a service.

2.	Working and Implementation of Software as a service.
3.	Working and Implementation of Platform as a services
4.	Practical Implementation of File sharing and Storage as a Service
5.	Create Google form for accepts details of student and create test page and generate result
6.	Working and Implementation of identity management.
7.	Write a program for web feed.
8.	Demonstration and implementation of cloud on single sign on.
9.	Practical Implementation of cloud security.
10.	Installing and Developing Application Using Google App Engine.
11.	Implement VMWAreESXi Server
12.	Managing and working of cloud xen server.
13.	Working with Aneka and demonstrate how to Managing cloud computing Resources.
14.	Create a Virtual Machine using Virtual Box.
15.	Create and host static web page using any cloud provider.
16.	Demonstrate how to managing cloud computing Resources.
17.	Using OpenNebula to manage heterogeneous distributed data centre Infrastructure.
Examination Scheme	**CIE : 15 Marks** **SEE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-514-MJ-TH	Major Elective	C# .NET Programming	TH	2	2

Objectives

- To understand the DOTNET framework
- Develop deep understanding of C# language features
- Build strong concepts of OOP's and implement the same in C#.
- To understand the concept of multi-threading & files
- To understand and implement the controls & properties of Windows forms
- To develop database centric applications using ADO.NET.

Course Outcomes

On Completion of this course, student will be able to -

CO1:Understand the features of Dot Net Framework along with the features of C#

CO2: Interpret and Develop Interfaces for real-time applications.

CO3: Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.

CO4: Design & Implement the application using multithreading & File handling

CO5: Design and Implement Windows Application using Windows Forms & tools application using Database in C#

CO6: Design and Implement Custom Application Using Windows Form & ADO.NET in C#

Course Contents:

Chapter-1	**Introduction to .Net Framework**	**Hours: 02**

1.1 Overview of .NET framework &.Net Architecture The Common Language Runtime (CLR) Microsoft Intermediate Language (MSIL) Code, Just In Time Compilers (JITers), The Framework Class Library (FCL), The Common Languages Specification (CLS), The Common Type System (CTS), Garbage Collection (GC),		
Chapter-2	**Introduction to C#.Net**	**Hours: 04**
2.1 Basics of C#. Language (Console Application) Namespace, Variables and Expressions, Type Conversion Boxing and Un-boxing Flow Control Functions Debugging and error handling **2.2** Array One-dimensional & two-dimensional array 2.3 Exception handling System Defined and User Defined		
Chapter-3	**OOPS with C#**	**Hours: 05**
3.1 Object Oriented Concept 3.2 Object and Classes 3.3 Class properties: Access modifiers, Implementation of class 3.4 Constructor, 3.5 Inheritance 3.6 Polymorphism & Interface 3.7 Abstract Class **3.8** Delegates **3.9** Multicasting & Anonymous Methods		
Chapter-4	**Data Structure**	**Hours: 02**
4.1 ArrayList 4.2 Collection 4.3 Dictionary 4.4 Hash Table		
Chapter-5	**Multithreading I/O Stream**	**Hours: 03**
5.1 Stream Reader, Stream Writer 5.2 File Mode 5.3 Opening & Closing File 5.4 Random Access File		
Chapter-6	**Assembly Components**	**Hours: 02**
6.1 .NET Assembly features 6.2 Structure of Assemblies 6.3 Calling assemblies, private and shared assemblies		

Chapter-7	Windows Programming	Hours: 06
7.1 Windows Forms Menus and Tool Bars, SDI and MDI applications, Building MDI applications. 7.2 Basic Controls Button, TextBox, Label, RadioButton, CheckBoxDateTimePicker, Timer ,PictureBox, ComboBox, ListBox, RichTextBox, MonthCalender 7.3 Container & Dialog Control GroupBox, Panel, Common Dialog boxes, ProgressBar		
Chapter-8	**Database Connectivity using ADO.NET**	**Hours: 06**
8.1 ADO.NET Architecture 8.2 Connection object, Commend Object 8.3 Dataset, DataReader & DataAdapter 8.4 SQL Commands (Insert, Delete,Update,Select) 8.5 Accessing Data with ADO.NET 8.6 Datagridview Data Binding: Insert, Update, Delete records		
Reference Books: 1. Programming in C#, E.Balagurusamy, 2. Professional C# ,Wrox Publication 3. C# The Complete Reference", Shildt, TMH 4. Database Programming with C#, By Carsten Thomsen, Apress **Web Reference :-** 1. Free Online Courses on Udemy Basics of Object Oriented Programming with C# , 2. Getting Started with C# Free Online Video - https://dotnet.microsoft.com/en-us/learn/csharp		
Examination Scheme	**CIE : 15 Marks** **SEE : 35 Marks**	

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	**CS-515-MJ-PR**	**Major Elective**	**Lab Course on CS-514-MJ-TH**	**PR**	**2**	**4**

Objectives

- To understand the DOTNET framework
- Develop deep understanding of C# language features
- Build strong concepts of OOP's and implement the same in C#.
- To understand the concept of multi-threading & files
- To understand and implement the controls & properties of Windows forms
- To Develop database centric applications using ADO.NET.

Course Outcomes

On Completion of this course, student will be able to -

CO1:Understand the features of Dot Net Framework along with the features of C#

CO2: Interpret and Develop Interfaces for real-time applications.

CO3: Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.

CO4: Design & Implement the application using multithreading & File handling

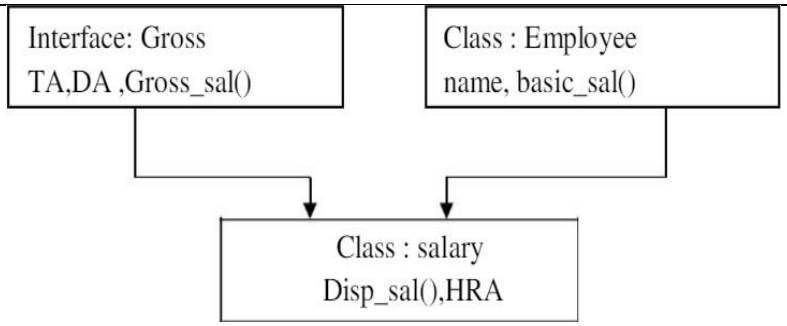
CO5: Design and Implement Windows Application using Windows Forms & tools application using

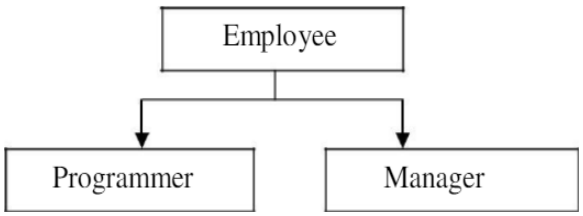
Database in C#

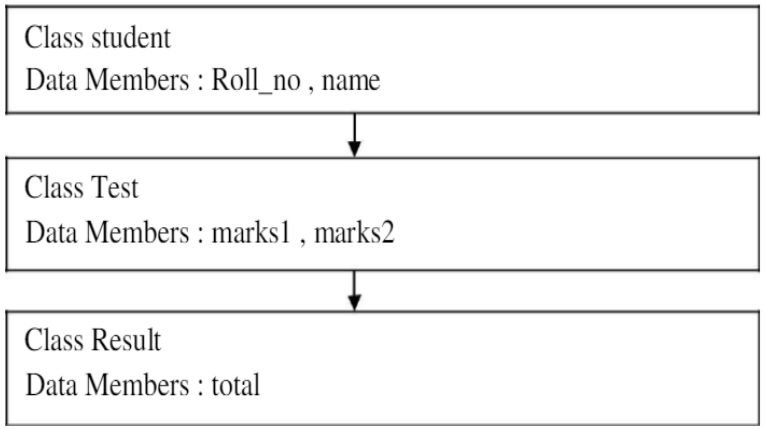
CO6: Design and Implement Custom Application Using Windows Form & ADO.NET in C#

Course Contents:

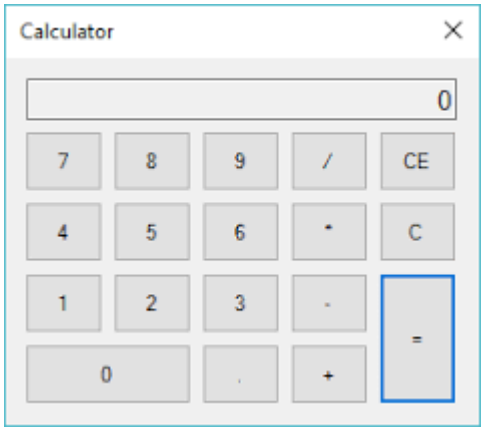
Assign No.	Practical Assignment
1 - 10	**C# Introduction** 1. Write a C# program to find the factorial of a given number. 2. Write a C# program to check whether a given number is prime or not. 3. Write a C# Sharp program to print on screen the output of adding, subtracting, multiplying and dividing of two numbers which will be entered by the user. 4. Write a C# program to check whether the given string is a palindrome or not 5. Write a C# program to find the second largest integer in an array using loop? 6. Write a C# program to sort an array in ascending and descending order. 7. Write a C# program to find minimum & maximum from array? 8. Write a C# program to create an MXN matrix and perform the following operation. a. Addition b. Multiplication c. Transpose 9. Write a C# program to create an MXN matrix and perform the following operation. a. Upper Triangular b. Lower Triangular c. Addition of row elements d. Addition of column elements e. Addition of diagonal elements 10. Write a C# program to accept one string &character , find the occurrence of char from string using function
11-19	**OOPs Concepts:** 11. Write a program to define a class Students having data members rollno, name. Accept data for 5 student‟s and display the name of student whose roll no is 3. 12. Write a program to swap three integer and three float numbers using the concept of Function overloading. 13. Implement a base class **Person**. Derive classes **Student** and **Instructor** from **Person**. A Person has aname and a birthday. A student has a batch, course and an Instructor has a salary. Write the class definitions, the constructor and the member function print () for all classes. 14. C# program to demonstrate the example of multilevel inheritance. 15. Write an application that receives the following information from a set of students: Student Id: Student Name: Course Name: Date of Birth: The application should also display the information of all the students once the data is Entered. 16. Write a program to declare class Distance having data members dist1, dist2, dist3. Initialize the two data members using constructor and store their addition in third data member using function and display addition. 17. Program to implement the following multiple inheritance using interface.

<table>
<tr><td></td><td>


18. Write a program for above class hierarchy for the Employee where the base class is Employee and derived class and Programmer and Manager. Here make display function virtual which is common for all and which will display information of Programmer and Manager interactively.


19. Write a program to implement multilevel inheritance from the following figure. Accept and display data for one student.


</td></tr>
<tr><td>20-21</td><td>Data Structure
20. Write a C# program to implement a stack with push and pop operations. Find the top element of the stack and check if the stack is empty or not.
21. Write a C# program to find the top and bottom elements of a given stack.</td></tr>
<tr><td>22-27</td><td>Multithreading and I/O Stream
22. C# program to assign the name to the thread
23. C# program to demonstrate the concept of parameter passing for thread
24. C# program to read data from file character by character till the end of the file
25. C# program to compare the content of two files using StreamReader class
26. C# program to get the size of a specified folder including sub-folder
27. C# program to demonstrate the BinaryReader and BinaryWriterclasses</td></tr>
<tr><td>28-30</td><td>Assembly:
28. Write a C# program which will demonstrate use of private assembly.
29. Write a C# program which will demonstrate use of public assembly.
30. Write a C# program which will demonstrate use of shared assembly.</td></tr>
</table>

31-32	**Exception Handling:** 31. Write a C# program that reads a list of integers from the user. Handle the exception that occurs if the user enters a value outside the range of Int32 32. Write a C# program that prompts the user to input a numeric integer and throws an exception if the number is less than 0 or greater than 1000.
33-37	**Windows Programming** 33. Create a windows application to perform following basic arithmetic operations  34. Create an application that accepts a number from a user in the textbox named num‟. Check whether the number in the textbox num‟ is palindrome or not. Print the message accordingly in the label control named lbldisplay when the user clicks on the button check.
	35. Create an application which will ask the user to input his name and a message, display the two items concatenated in a label, and change the format of the label using radio buttons and checkboxes for selection , the user can make the label text bold ,underlined or italic and change its color . include buttons to display the message in the label, clear the text boxes and label and exit. 36. Create a user control that contains a list of colors. Add a button to the Form or testbox which when clicked changes the color of the Form or textbox to the color selected from the list. 37. Create a RadioButtonList that displays the names of some flowers in two columns. Bind a label to the RadioButtonList so that when the user selects an option from the list and clicks on a button, the label displays the flower selected by the user.
38-42	**Database Connectivity using ADO.Net:** 38. Write a C# application using ADO.NET to verify if the connection is established with the database or not. Display appropriate messages 39. Write a C# application using ADO.NET to perform insert, delete, update and select operation. 40. Create table Student with the following columns and datatypes. Student (rollnoInt, Name Char(20), DOB Date) Insert few records into the table. Change the candidate name from „Ram‟ to „Krishnan‟. Drop the table. Display all the records in gridview. 41. Create table Employee with the following columns and datatypes & perform the following operation

	i. Display all the employees whose SAL is less than 3000. ii. Display all the employees who are working as MANAGER or ANALYST. iii. Select all the employees who work in department 20 and whose salary exceeds 2000. iv. Select the details of employees whose name starts with „J". v. Update the salary of employees by 1000 for those drawing less than 2000. vi. Find out the average salaries of employees department wise. 42. Create a table "students" with the below given column. Insert records in that & perform the following operation. i. Delete those students who get less than 40 marks. ii. Display those students name who get more than 90% iii. Display the name of students' whose name starts with _.
Examination Scheme	**CIE : 15 Marks** **SEE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Faculty of Science and Technology**
Program	**M.Sc. (Computer Science)**
Class	**F.Y M.Sc. (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CS-531-RM-TH	RM	Research Methodology	TH	4	4

Objectives

- Research Methodology course are designed to equip students with the necessary knowledge, skills, and understanding of various research techniques and methodologies.
- Students should be familiar with various data collection techniques, such as surveys, interviews, observations, and experiments, and understand their strengths and limitations.
- Students should be aware of ethical considerations in research, including issues related to participant consent, privacy, confidentiality, and avoiding plagiarism.
- Its aim is to enable students to conduct research effectively, critically evaluate existing research, and contribute to the advancement of knowledge in their respective fields.

Course Outcomes

On Completion of this course, student will be able to -

CO 1. Understand of the fundamental concepts of research, including the research process, research questions, hypotheses, and variables.

CO 2. Conduct a comprehensive literature review to identify relevant studies, synthesize existing knowledge, and identify research gaps.

CO 3. Identify research problems, formulate research questions, and design appropriate methodologies to address these problems

CO 4. Identify and select appropriate research designs, such as experimental, observational, survey,

qualitative, or mixed-methods, based on the research objectives.

CO 5. Apply appropriate data analysis methods, including statistical techniques or qualitative analysis, to draw meaningful conclusions from research data.

CO 6. Develop a well-structured research proposal, outlining research questions, methodology, expected outcomes, and a rationale for the study.

CO 7. Communicate research findings effectively through written reports, presentations, and academic papers.

CO 8. Gain an appreciation for the importance of research in contributing to the advancement of knowledge in their field of study and broader society.

CO 9. Understand the principles of research ethics and integrity and apply them in their research.

Course Contents:

Chapter-1	**Introduction to Research Methodology**	**Hours: 10**

1.1 Meaning of Research
1.2 Objectives of Research
1.3 Motivation in Research
1.4 Types of Research
1.5 Research Approaches
1.6 Significance of Research
1.7 Researcher and Characteristics of Researcher
1.8 Research Ethics and Integrity
1.9 Plagiarism and types of plagiarism
1.10 Introduction to Plagiarism check tools
1.11 Research Methods versus Methodology
1.12 Research and Scientific Method
1.13 Importance of Knowing How Research is Done
1.14 Criteria of Good Research

Chapter-2	**Literature Review and Formulation of Research Problems**	**Hours: 06**

2.1 Research Process
2.2 Reviewing the literature: purpose of a literature review
2.3 Literature resources
2.4 The Internet and a literature review
2.5 The Internet and research strategies and methods
2.6 Conducting and Evaluating literature reviews
2.7 Formulation of research problem
2.7.1 What is a Research Problem?
2.7.2 Selecting the Problem
2.7.3 Necessity of Defining the Problem
2.7.4 Technique Involved in Defining a Problem

Chapter-3	**Research Design**	**Hours: 08**

3.1 Meaning of Research Design
3.2 Need for Research Design
3.3 Features of a Good Design
3.4 Important Concepts Relating to Research Design
3.5 Different Research Designs/Methods
3.5.1 Pure and Applied Research
3.5.2 Exploratory or Formulative Research
3.5.3 Descriptive Research
3.5.4 Diagnostic Research
3.5.5 Evaluation Studies
3.5.6 Action Research
3.5.7 Experimental Research
3.5.8 Analytical Study or Statistical Method
3.5.9 Historical Research
3.5.10 Surveys
3.5.11 Case Study
3.5.12 Field Studies

Chapter-4	**Hypothesis and Sampling**	**Hours: 10**

4.1 What is Hypothesis?
4.2 Nature & Characteristics of Hypothesis
4.3 Significance of Hypothesis
4.4 Types of Hypothesis
4.5 Sources of Hypothesis
4.6 Characteristics of Good Hypothesis
4.7 What is sampling?
4.8 Aims of Sampling
4.9 Characteristics of Good Sample
4.10 Basis of Sampling
4.11 Merits and demerits of Sampling
4.12 Sampling Techniques or Methods
4.13 Probability Sampling Methods
4.14 Non-Probability Sampling Methods
4.15 Sample Design and Choice of Sampling Technique

Chapter-5	**Data Collection, Processing and Analysis of Data**	**Hours: 10**

5.1 Collection of Primary Data
5.2 Method of data Collections - Observation, Interview, Questionnaires and Schedules
5.3 Difference between Questionnaires and Schedules
5.4 Some Other Methods of Data Collection
5.5 Collection of Secondary Data
5.6 Selection of Appropriate Method for Data Collection
5.7 Case Study Method

5.8 Processing Operations and Some Problems in Processing
5.9 Elements/Types of Data Analysis
5.10 Statistics in Research
5.11 Measures of Central Tendency, Dispersion, Asymmetry (Skewness)
5.12 Measures of Relationship - Chi-Square, t-test, ANNOVA(f-test),Z-test
5.13 Simple Regression Analysis, and Multiple Correlation and Regression
5.14 Partial Correlation and Association in Case of Attributes
5.15 Quantitative and Qualitative Data Analysis Tools

Chapter-6	**Interpretation and Report Writing**	**Hours: 08**

6.1 Meaning of Interpretation, Why Interpretation?
6.2 Technique of Interpretation
6.3 Precaution in Interpretation
6.4 Significance of Report Writing
6.5 Different Steps in Writing Report
6.6 Layout of the Research Report
6.7 Types of Reports (Research Proposal/Synopsis, Research Paper, and Thesis)
6.8 Oral Presentation
6.9 Mechanics of Writing a Research Report
6.10 Precautions for Writing Research Reports

Chapter-7	**Publication Ethics and Open Access Publishing**	**Hours: 08**

7.1 Publication ethics: definition, introduction and importance
7.2 Best practices/standards setting initiatives and guidelines: COPE, WAME, etc.
7.3 Conflicts of interest
7.4 Publication misconduct: definition, concept, problems that lead to unethical behaviour and vice versa, types
7.5 Violation of publication ethics, authorship and contributor ship
7.6 Identification of publication misconduct, complaints and appeals
7.7 Predatory publishers and journal
7.8 Open access publications and initiatives
7.9 SHERPA/RoMEO online resource to check publisher copyright & self-archiving policies
7.10 Software tool to identify predatory publications developed by SPPU
7.11 Journal finder/ journal suggestion tools viz. JANE, Elsevier Journal Finder, Springer Journal Suggester, etc.
7.12 E-Resources for research: Google Scholar, Shodh Ganaga, Shodh Gangotri

Reference Books:

1. Researching Information Systems and Computing by Briony J Oates, SAGE SOUTH ASIA Ed
2. Research Methodology: A Step-by-Step Guide for Beginners, Kumar, Pearson Education.
3. Research Methodology Methods and Techniques, Kothari, C. R., Wiley Eastern Ltd.
4. The Research Methods Knowledge Base, by William M. K. Trochim, James P. Donnelly
5. Introducing Research Methodology: A Beginner‟s Guide to Doing a Research Project, Uwe Flick

<table>
<tr><td colspan="2">6. A Guide to Research and Publication Ethics by Partha Pratim Ray, New Delhi Publishers
7. RESEARCH & PUBLICATION ETHICS by Wakil kumar Yadav, NOTION PRESS
8. Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd.</td></tr>
<tr><td>Examination Scheme</td><td>CIE : 30 Marks
SEE : 70 Marks</td></tr>
</table>

The Poona Gujarati Kelavani Mandal's
HARIBHAI V. DESAI COLLEGE
of Arts, Science & Commerce (Autonomous)
Affiliated to Savitribai Phule Pune University
(Linguistic Minority Institution) AICTE NO. : 1-44457797714
ID No.: PU / PN / ASC / 057/ (1984)
NAAC Grade B++ (2.86 CGPA) ■ AISHE CODE : C-41829
Principal:
Dr. Rajendra G. Gurao
M.Sc., Ph.D.
Email: principal@hvdesaicollege.edu.in

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-551-MJ-TH	Major Core	Design and Analysis of Algorithms	Theory	4	4

Course Objectives:

- *To design the algorithms*
- *To Understand different design strategies and the use of data structures in improving algorithm performance*
- *To critically analyze the efficiency of alternative algorithmic*
- *To develop the ability to understand and design algorithms in the context of space and time*
- *complexity*

Course Outcomes:

- *CO1:Analyze worst-case running times of algorithms using asymptotic analysis.*
- *CO2:Compare between different data structures. Pick an appropriate data structure for a design situation.*
- *CO3:Ability to design algorithms using standard paradigms like:Greedy, Divide and Conquer, Dynamic Programming and Backtracking.*
- *CO4:Able to Explain the major graph algorithms and Employ graphs to model engineering problems, when appropriate.*
- *CO5:Able to Compare between different data structures and pick an appropriate data structure for a design situation.*

Course Contents:

Chapter-1	Chapter Name: Basics of Algorithms	Hours: 06

1.1 Algorithm definition and characteristics
1.2 Space complexity
1.3 Time complexity- worst case, best case, average case
1.4 Complexity, asymptotic notation
1.5 Recursive and non-recursive algorithms

1.6 Sorting algorithms : insertion sort, heap sort, bubble sort 1.7 Sorting in linear time: counting sort, concept of bucket and radix sort 1.8 Searching algorithms: Linear, Binary		
Chapter-2	**Chapter Name:** Divide and Conquer strategy	**Hours: 10**
2.1 General method, control abstraction 2.2 Binary search 2.3 Merge sort, Quick sort 2.4 Comparison between Traditional Method of Matrix Multiplication vs. Strassen‟s Matrix Multiplication 2.5 Writing simple algorithm using Divide and conquer strategy: power(x,n), find occurrence of a number from array of N integers, to find minimum from an array, minimax algorithm, largest number multiplication, simple convex algorithm		
Chapter-3	**Chapter Name:** Greedy Method	**Hours: 10**
3.1 Knapsack problem 3.2 Job sequencing with deadlines 3.3 Minimum-cost spanning trees: Kruskal and Prim‟s algorithm 3.4 Optimal merge patterns 3.5 Huffman coding 3.6 Shortest Path :Dijkstra‟s Algorithm		
Chapter-4	**Chapter Name:** Dynamic Programming	**Hours: 12**
4.1 Principle of optimality 4.2 Matrix chain multiplication 4.3 0/1 Knapsack Problem i)Merge & Purge ii)Functional Method 4.4 Bellman Ford Algorithm 4.5 Coin changing problem 4.6 Travelling Salesperson problem 4.7 Longest common subsequence 4.8 String editing		
Chapter-5	**Chapter Name:** Decrease and Conquer	**Hours: 06**
5.1 Definition of Graph Representation of Graph 5.2 By Constant - DFS and BFS 5.3 Topological sorting 5.4 Articulation Point and Bridge edge		
Chapter-6	**Chapter Name:** Backtracking	**Hours: 07**
6.1 General method 6.2 Fixed Tuple vs. Variable Tuple Formulation 6.3 n- Queen‟s problem 6.4 Graph colouring problem 6.5 Hamiltonian cycle 6.6 Sum of subsets		
Chapter-7	**Chapter Name:** Branch and Bound Technique	**Hours: 06**
7.1 Introduction : Branch and bound terms like definition of live node, E-node, Dead node, Least cost (LC) search, Least cost Branch and Bound (LCBB) 7.2 0/1 knapsack problem using LCBB method (fixed tuple size) 7.3 Travelling Salesman problem using LCBB method (variable tuple size)		
Chapter-8	**Chapter Name:** Problem Classification	**Hours: 03**
8.1 The class of P, NP, NP-hard and NP -Complete 8.2 Relationship among P class, NP class, NP-hard and NP -Complete		
Reference Books:		

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI Learning Private Limited, 2012
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Data Structures and Algorithms, Pearson Education, Reprint 2006.
3. Harsh Bhasin, Algorithms Design and Analysis, Oxford university press, 2016.
4. S. Sridhar, Design and Analysis of Algorithms, Oxford university press, 2014.

Web References:

www.w3schools.com
www.tutorialspoint.com
www.javatpoint.com
www.geeksforgeeks.com
www.programiz.com
www.theserverside.com
www.educba.com
www.sanfoundry.com
www.prepbytes.com
www.codercampus.com

Examination Scheme: **IE : 30 Marks** **EE : 70 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-552-MJ-TH	Major Core	Mobile App Development Technologies	Theory	4	4

Course Objectives:

- Students should learn the Android Fundamentals and Android architecture framework.
- Students should understand GUI Design concepts and design Android GUI Layout.
- Students should be able to design visually appealing and intuitive user interfaces for Android apps, using appropriate layouts, widgets, and styles.
- Students should be Develop and design event-driven programming with UI Controls.
- Students should understand how to manage data in Android applications, including using SQLite databases, shared preferences, and data storage.
- Students should develop problem-solving skills related to Android app development, addressing challenges in app design and implementation.
- Students should understand the Phone Gap Programming.

Course Outcomes:

- *CO 1. To provide students with a solid understanding of the mobile app development, Android operating system, its architecture, components, and the software development kit (SDK).*
- *CO 2. To teach students how to build Android applications from scratch, including UI design, handling user interactions, and integrating various features.*
- *CO 3. To learn about Android's UI components, layouts, and design principles to create visually appealing and user-friendly interfaces.*
- *CO 4. To know various methods of data storage in Android applications, such as using SQLite databases, shared preferences, and cloud-based solutions.*
- *CO 5. To empower students to independently design, develop, and deploy their Android applications using advanced android tools.*

- *CO 6. To understand how to utilize built-in sensors and hardware components on Android devices, such as, Bluetooth, WiFi, Media Player and Camera, in their applications.*
- *CO 7. To Get knowledge of Phone Gap Programming*

Course Contents:

Chapter-1	**Chapter Name:** Introduction Mobile Technologies	**Hours: 03**

1.1. Introduction to Mobile Computing- Features, Advantages, Disadvantages and Applications
1.2. Factors in Developing Mobile Applications
1.3. Mobile Apps and Types of Mobile Apps
1.4. Mobile Apps Design & Development Process
1.5. Mobile Operating System: IOS, BlackBery, Android, Windows Phone, PlamOS, SymbianOS, PhoneGap etc.

Chapter-2	**Chapter Name:** Fundamentals of Android Programming	**Hours: 06**

2.1. Introduction to Android - Overview and Evolution of Android , Features of Android,
2.2. Android Architecture
2.3. Android Environment Setup Android-SDK, Eclipse, Emulators /Android AVD
2.4. First Android Application.
2.5. Introduction to Components of an Android Application
2.6. Resources and Manifest File
2.7. Android App / Project Folder Structure

Chapter-3	**Chapter Name:** Android Activity, Intents, and Services	**Hours: 06**

3.1. Android Activity and Android Activity life Cycle
3.2. Toast in Android
3.3. Intents: Implicit, Explicit, and Intent Filters
3.4. Android Services and Service Life Cycle
3.5. Android Fragments

Chapter-4	**Chapter Name:** Android UI Layouts and Controls for GUI Design	**Hours: 12**

4.1. Android View, View Groups- Linear Layout, Relative Layout, Table Layout, Frame Layout, Web View, List View, Grid View
4.2. Android UI Controls – TextView, EditText, AutoCompleteTextView, Button, ImageButton, ToggleButton, CheckBox, RadioButton, RadioGroup, ProgressBar, Spinner,TimePicker, DatePicker, SeekBar, AlertDialog, Switch, RatingBar
4.3. Event-driven Programming in Android, List and Adaptors
4.4. Android Styles and Themes

Chapter-5	**Chapter Name:** Android Menus, Threads, Notification and Alarms	**Hours: 08**

5.1. Creating a splash screen, Threads in Android,
5.2. Threads running on UI thread (runOnUiThread),
5.3. Worker thread, Handlers & Runnable, AsynTask (in detail)
5.4. Android Menus - Options, Context, Popup
5.5. Android Notification- Progress and Push
5.6. Android Alarms

Chapter-6	**Chapter Name:** Android ContentProviders, Broadcast Receivers and Parsing	**Hours: 08**

6.1. Basic operation of SQLite Database, Android Application Priorities
6.2. Android Content Providers – SQLite Programming : Open Helper and create the database, open and close a database, and insert, update, and delete operation in database
6.3. Android BroadcastReceivers
6.4. Android Parsing- JSON, and XML

Chapter-7	**Chapter Name:** Advanced Android Programming	**Hours: 09**

7.1. Accessing Phone Service (Call, SMS, MMS), Android Email 7.2. Location-based services 7.3. Storage in Android-Shared Preferences, Internal and External Storage 7.4. Multimedia in Android – Android Camera, Audio Player. Video player 7.5. Android Bluetooth, Android WiFi		
Chapter-8	**Chapter Name:** Phone Gap Programming	**Hours: 08**
8.1. Why Use Phone Gap? 8.2. How Phone Gap Works, designing for the Container, writing 8.3. Phone Gap Applications, Building Phone Gap Applications, 8.4. Phone Gap Limitations, Phone Gap Plug-Ins 8.5. Hello, World! Program		
Reference Books: 1. Professional Android 2 Application Development by Reto Meier, Wiley India Pvt Ltd publication. 2. Android Cookbook by Ian F. Darwin O‟Reilly Media, Inc. 3. Beginning Android by Mark L. Murphy, Wiley India Pvt Ltd publication. 4. Professional Android by Sayed Y Hashimi and Satya Komatineni, Wiley India Pvt Ltd publication. 5. Building Android Apps by in easy Steps, McGraw-Hill Education publication. 6. 20 Recipes for Programming PhoneGap: Cross-Platform Mobile Development for Android and iPhone by Jamie Munro O'Reilly Media 7. PhoneGap Beginner's Guide - Andrew Lunny Packt Publishing		
Examination Scheme: **IE : 30 Marks** **EE : 70 Marks**		

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-553-MJ-TH	Major Core	Software Project Management	Theory	2	2

Course Objectives:

- *To get skills that are required to ensure successful medium and large scale software projects*
- *To study Requirements Elicitation, Project Management, Verification &Validation and Management of Large Software Engineering Projects.*
- *To learn to select and apply project management techniques for process modeling, planning, estimation, process metrics and risk management*

Course Outcomes:

- *CO1: Learn the skills that are required to ensure successful medium and large scale software projects.*
- *CO2: Examine Requirements Elicitation, Project Management, Verification &Validation and Management of Large Software Engineering Projects.*
- *CO3: Get knowledge to select and apply project management techniques for process modeling, planning, estimation, process metrics and risk management.*
- *CO4: Understand the concepts, skills, tools, and techniques of software project management.*

Course Contents:

Chapter-1	**Chapter Name: Introduction to Project Management**	**Hours: 4**

1.1 What is a Project?
1.2 What is Project management?
1.3 Project phases and project life cycle
1.4 Organizational structure
1.5 Qualities of Project Manager
1.6 Work Breakdown Structure
1.7 Need for Software Project Management

Chapter-2	**Chapter Name: Project Management Components**	**Hours: 4**

2.1 Project Integration Management-Project plan 2.2 development and execution 2.3 Change controls and CCB 2.4 Configuration management		
Chapter-3	**Chapter Name: Scope, Time and Cost Management**	**Hours: 6**
3.1 Strategic planning 3.2 Scope planning, definition 3.3 Verification and control 3.4 Activity planning 3.5 Schedule development and control 3.6 GANTT Chart 3.7 Basic cost concept 3.8 Cost estimation and Control 3.9 COCOMO model 3.10 BASIC COCOMO NUMERICALS		
Chapter-4	**Chapter Name:Quality Management and Quality Standards**	**Hours: 4**
4.1 Quality planning and assurance 4.2 CMM levels 4.3 KPA‟s 4.4 PSP/TSP 4.5 Six Sigma		
Chapter-5	**Chapter Name:Human Resource Management and Communication Management**	**Hours: 4**
5.1 Staff acquisition 5.2 Information distribution 5.3 Reporting		
Chapter-6	**Chapter Name:Risk and Procurement Management**	**Hours: 4**
6.1 Risk identification 6.2 Quantification and control 6.3 Contract administration		
Chapter-7	**Chapter Name:Stakeholder Management and Software Metrics**	**Hours: 4**
7.1 Identifying Stakeholders 7.2 Planning, Managing and Monitoring Stakeholder Engagement 7.3 The scope of software metrics 7.4 Size- oriented metrics 7.5 Function oriented 7.6 Software metrics data collection		

Reference Books:

1. The Software Development Project: Planning and Management by Phillip Bruce and Sam M Pederson
2. Software Project Management : A Process-Driven Approach by Ashfaque Ahmed
3. Software Engineering Project Management by Richard Thayer, Edward Yourdon WILEY.
4. Introduction to Software Project Management by Adolfo Villafiorita CRC Press
5. Software Engineering by Roger Pressman McGraw-Hill
6. Software Metrics for Project Management and process improvement by Robert B. Grady Prentice hill

Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-554-MJ-PR	Major Core	Lab course on CS-551-MJ-TH	Practical	2	4

Course Objectives:

- To design the algorithms
- To learn basic Algorithm Analysis techniques and understand the use of asymptotic notation
- To Understand different design strategies
- To Understand the use of data structures in improving algorithm performance
- To critically analyze the efficiency of alternative algorithmic
- To understand different algorithm design techniques.
- To provide foundation in algorithm design and analysis
- To develop the ability to understand and design algorithms in the context of space and time complexity

Course Outcomes:

On Completion of this course, student will be able to -
CO1: Analyze worst-case running times of algorithms using asymptotic analysis.
CO2: Compare between different data structures. Pick an appropriate data structure for a design situation.
CO3: Ability to design algorithms using standard paradigms like: Greedy, Divide and Conquer, Dynamic Programming and Backtracking.
CO4: Able to Explain the major graph algorithms and Employ graphs to model engineering problems, when appropriate.
CO5: Able to Compare between different data structures and pick an appropriate data structure for a design situation.

Course Contents:

Practical Assignments

1. Write programs in C/C++/ Java to sort a list of n numbers in ascending order using heap sort.

2. Write a program in C/C++/ Java to sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted. The elements can be read from a file or can be generated using the random number generator.
3. Write a program in C/C++/ Java to implement a Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted. The elements can be read from a file or can be generated using the random number generator.
4. Write a program in C/C++/ Java to implement Strassen‟s Matrix multiplication
5. Write a program in C/C++/ Java to find Minimum Cost Spanning Tree of a given undirected graph using Kruskal‟s algorithm
6. Write a program in C/C++/ Java to find Minimum Cost Spanning Tree of a given undirected graph using Prim‟s algorithm
7. Write a program in C/C++/ Java to from a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijikstra‟s algorithm
8. Write a program in C/C++/ Java to implement Knapsack problems using Greedy method
9. Write a program in C/C++/ Java to implement optimal binary search tree and also calculate the best case and worst case complexity.
10. Write a program in C/C++/ Java to implement huffman Code using greedy methods and also calculate the best case and worst case complexity.
11. Write a program in C/C++/ Java to find Minimum number of multiplications in Matrix Chain Multiplication
12. Write a Program in C/C++/Java to find only length of Longest Common Subsequence.
13. Write programs in C/C++/ Java to implement DFS and BFS. Compare the time complexity
14. Write a program in C/C++/ Java for finding Topological sorting for Directed Acyclic Graph (DAG)
15. Write a program in C/C++/ Java to determine if a given graph is a Hamiltonian cycle or not
16. Write a Java Program in C/C++/ Java to implement Traveling Salesman Problem using nearest neighbor algorithm
17. Write a program in C/C++/ Java a to implement Graph Coloring Algorithm
18. Write a program in C/C++/ Java to implement Sum of Subset by Backtracking
19. Write a program in C/C++/ Java to solve N Queens Problem using Backtracking
20. Write a program in C/C++/ Java to solve 4 Queens Problem using Backtracking
21. Write a program in C/C++/ Java to find out longest common subsequence from the given strings
22. Write a program in C/C++/ Java to find out live node, E node and dead node from a given graph
23. Write a program in C/C++/ Java to find out solution for travelling salesman problem using LCBB from a given matrix.
24. Write a program in C/C++/ Java to find out solution for 0/1 knapsack problem

Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-555-MJ-PR	Major Core	Lab course on CS-552-MJ-TH	Practical	2	4

Course Objectives:

- Identify and understand the concepts of open-source mobile technology.
- Understand the Android architecture framework.
- Understand GUI Design concepts and design Android GUI Layout.
- Develop and design event-driven programming with menus and dialog boxes.
- Design and develop applications with databases.

Course Outcomes:

On Completion of this course, student will be able to -

- *CO 8. To teach students how to build Android applications from scratch, including UI design, handling user interactions, and integrating various features.*
- *CO 9. To learn about Android's UI components, layouts, and design principles to create visually appealing and user-friendly interfaces.*
- *CO 10.To empower students to independently design, develop, and deploy their Android applications using advanced android tools.*

Course Contents:

Practical Assignments

1. Java Android Program to demonstrate login form with validation.
2. Java Android Program to demonstrate Registration form with validation.
3. Create the simple calculator and perform appropriate operation
4. Create an Android application which examine, that a phone number, which a user has entered is in the given format. * Area code should be one of the following: 040, 041, 050, 0400, 044 * There should 6- 8 numbers in telephone number (+ area code).
5. By using Spinner, Buttons. Write a program to draw GUI.
6. Create an Android application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts and shows to the user how many answers were right and shows the result to user.
7. Construct an app to display the image on date wise.
8. Construct image switcher using setFactory().

9. Construct a bank app to display different menu like windrow, deposite etc.

10. Create an Android application, where the user can enter player name and points in one view and display it in another view.

11. Create an Android application, the user can enter 10 students information and stored it in file and display student information in second view and also search the particular student information.

12. Write an application to accept two numbers from the user, and displays them, but reject input if both numbers are greater than 10 and asks for two new numbers.

13. Create table Customer (id, name, address, phno). Create Application for Performing the following operation on the table. (using sqlite database) i) Insert New Customer Details. ii) Show All the Customer Details

14. Create an application that allows the user to enter a number in the textbox named „getnum". Check whether the number in the textbox „getnum" is palindrome or not. Print the message accordingly in the label control named lbldisplay when the user clicks on the button „check".

15. Create Following Table: Emp (emp_no,emp_name,address,phone,salary) Dept (dept_no,dept_name,location) Emp-Dept is related with one-many relationship. Create application for performing the following Operation on the table 1) Add Records into Emp and Dept table. 2) Accept Department name from User and delete employee information which belongs to that department.

16. Java Andorid Program to Perform all arithmetic Operations using Calculators

17. Java Android Program to Change the Image Displayed on the Screen

18. Java Android Program to Demonstrate Alert Dialog Box

19. Java Android Program to Demonstrate the Menu Application

20. Java Android Program to Demonstrate List View Activity with all operations (Insert, delete, Search).

21. Java Android Program to Display SMS from the Phone Numbers, which are in Your Contacts

22. Java Android Program to send email with attachment.

23. Create an Android application which will ask the user to input his name and a message, display the two items concatenated in a label, and change the format of the label using radio buttons and check boxes for selection, the user can make the label text bold, underlined or italic and change its color .include buttons to display the message in the label, clear the text boxes and label and then exit.

24. Write a program to search a specific location on Google Map.

25. Write a program to perform Zoom In, Zoom Out operation and display Satellite view,Terrain view of current location on Google Map.

26. Create Simple PhoneGap Application to show "Hello World" Message.

Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-560-MJ-TH	Major Elective	Full Stack Development - I	Theory	2	2

Course Objectives:

- *Get familiar with the MEAN stack*
- *Learn advanced ES6 features in Javascript & typescript*
- *Learn front end development using Angular*
- *Create backend APIs using NodeJS and ExpressJS*
- *Develop full stack application using MEAN stack*
- *Learn how to secure & scale MEAN stack applications Deploy MEAN stack application on production/local server*

Course Outcomes:

On Completion of this course, student will be able to -
CO1: Learn about the benefits of using MEAN stack and how to install and configure it
CO2: Learn advanced ES6 features in JavaScript and Typescript
CO3: Learn about Angular architecture, components, directives, pipes, forms, routing, and services.
CO4: Learn about the event loop, asynchronous programming, modules, packages, and streams.
CO5: Learn about the MVC pattern, routing, HTTP requests and responses, middleware, and error handling.
CO6: Create a full-stack MEAN stack application and deploy it to a production/local server.

Course Contents:

Chapter-1	**Introduction to MEAN Stack**	**Hours: 2**

1.1 What is MEAN stack?
1.2 The benefits of using MEAN stack
1.3 The different technologies that make up MEAN stack
1.4 Installing and configuring the MEAN stack

Chapter-2	**Advanced ES6 features in JavaScript and Typescript**	**Hours: 8**

2.1 Introduction to ES6
2.1.1 let and const
2.1.2 Arrow functions

2.1.3 Template literals
2.1.4 destructuring assignment
2.1.5 Spread syntax
2.1.6 Modules/Classes
2.1.7 symbols
2.1.8 iterators/generators
2.1.9 map/set

2.2 Functional programming
2.2.1 Pure functions
2.2.2 Higher-order functions
2.2.3 Currying
2.2.4 Immutable data structures

2.3 Asynchronous programming
2.3.1 Promises
2.3.2 Async/await
2.3.3 Callbacks
2.3.4 Generators

2.4 TypeScript
2.4.1 What is TypeScript?
2.4.2 Benefits of using TypeScript
2.4.3 Installing TypeScript
2.4.4 Writing TypeScript code
2.4.5 Types in TypeScript Basic types, Enums, Interfaces, Classes, Generics

2.5 Advanced TypeScript
2.5.1 Modules
2.5.2 Decorators
2.5.3 Type narrowing
2.5.4 Type guards

Chapter-3	**AngularJS**	**Hours: 5**

3.1 Introduction to AngularJS
3.2 Angular architecture
3.3 Components, directives, and pipes
3.4 Forms and validation
3.5 Routing
3.6 Services
3.7 Introduction to RxJS library
3.8 Introduction to NgRx

Chapter-4	**Node.js**	**Hours: 5**

4.1 Introduction to Node.js
4.2 Event loop
4.3 Asynchronous programming
4.4 Modules
4.5 Packages
4.6 Streams

Chapter-5	**ExpressJS**	**Hours: 5**

5.1 Introduction to ExpressJS
5.2 The MVC pattern
5.3 Routing
5.4 HTTP requests and responses
5.5 Middleware
5.6 Error handling

Chapter-6	Building a MEAN Stack Application	Hours: 5
6.1 Create a full-stack MEAN stack application 6.2 Use all of the technologies learned in the course 6.3 Deploy the application to a production/local server		

Reference Books:

1. Beginning MEAN Stack by Greg Lim, Daniel Correa
2. Beginning Node.js, Express & MongoDB Development by Greg Lim
3. FULLSTACK Web Development by PANKAJ KAPOOR
4. Write Modern Web Apps With the Mean Stack by Jeff Dickey
5. Full Stack JavaScript Development With MEAN by Colin J Ihrig and Adam Bretz
6. Pro MEAN Stack Development by Elad Elrom
7. Web Application Development with MEAN by Amos Q. Haviv, Adrian Mejia, Robert Onodi
8. MEAN Cookbook: The meanest set of MEAN stack solutions around by Nicholas McClay
9. Node.js, MongoDB and Angular Web Development by Brad Dayley
10. MEAN Web Development by Amos Q. Haviv
11. Getting MEAN with Mongo, Express, Angular, and Node by Simon Holmes, Clive Herber
12. Full-Stack JavaScript Development by Eric Bush
13. Web Development with Node and Express by Ethen brown
14. JavaScript: The Good Parts by D Crockford
15. JavaScript - The Definitive Guide, 7th edition by David Flanagan
16. Effective TypeScript by Dan Vanderkam
17. Mastering TypeScript - Fourth Edition by Nathan Rozentals
18. Angular Development with TypeScript by Yakov Fain, Anton Moiseev
19. Express in Action by Evan Hahn
20. Node.js in Action by Mike Cantelon, Marc Harter, T.J. Holowaychuk, and Nathan Rajlich

Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-561-MJ-PR	Major Elective	Lab Course on CS-560-MJ-TH	Practical	2	4

Course Objectives:

- Understand Client-side Scripting Language
- Develop an AngularJS Single Page Application
- To Create and bind controllers with Javascript
- Apply filter in AngularJS application
- Understanding of the various components of a React application

Course Outcomes:

On Completion of this course, student will be able to -

- CO1: Describe appropriate uses for JavaScript and PHP
- CO2: Discuss, create, and debug semantically correct basic examples of dynamic web pages
- CO3: Construct individual components and entire applications using ReactJS
- CO4: Build an interactive web page using ReactJS

Course Contents:

Assign No.	Name of Practical Assignment
1	Create an HTML form that contain the Student Registration details and write a JavaScript to validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50.
2	Create an HTML form that contain the Employee Registration details and write a JavaScript to validate DOB, Joining Date, and Salary.
3	Create an HTML form for Login and write a JavaScript to validate email ID using Regular Expression.
4	Write angular JS by using ng-click Directive to display an alert message after clicking the element
5	Write an AngularJS script for addition of two numbers using ng-init, ng-model & ng-bind. And also Demonstrate ng-show, ng-disabled, ng-click directives on button component.
6	Using angular js display the 10 student details in Table format (using ng-repeat directive use Array to store data)
7	Using angular js Create a SPA that show Syllabus content of all subjects of MSC(CS) Sem II (use ng-view)

8	Using angular js create a SPA to accept the details such as name, mobile number, pincode and email address and make validation. Name should contain character only, mobile number should contain only 10 digit, Pincode should contain only 6 digit, email id should contain only one @, . Symbol
9	Using AngularJS create a SPA for Login System.
10	Create an HTML form using AngularJS that contain the Student Registration details and validate Student first and last name as it should not contain other than alphabets and age should be between 18 to 50 and display greeting message depending on current time using ng-show (e.g. Good Morning, Good Afternoon, etc.)(Use AJAX).
11	Create angular JS Application that show the current Date and Time of the System(Use Interval Service)
12	Using angular js create a SPA to carry out validation for a username entered in a textbox. If the textbox is blank, alert „Enter username‟. If the number of characters is less than three, alert ‟ Username is too short‟. If value entered is appropriate the print „Valid username‟ and password should be minimum 8 characters
13	Create an angular JS Application that shows the location of the current web page.
14	Create a Node.js file that will convert the output "Hello World!" into upper-case letters
15	Using nodejs create a web page to read two file names from user and append contents of first file into second file
16	Create a Node.js file that opens the requested file and returns the content to the client If anything goes wrong, throw a 404 error
17	Create a Node.js file that writes an HTML form, with an upload field
18	Create a Node.js file that demonstrate create database and table in MySQL
19	Create a node.js file that Select all records from the "customers" table, and display the result object on console
20	Create a node.js file that Insert Multiple Records in "student" table, and display the result object on console
21	Create a node.js file that Select all records from the "customers" table, and delete the specified record.
22	Create a Simple Web Server using node js
23	Using node js create a User Login System
24	Using node js create a eLearning System
25	Using node js create a Recipe Book
26	Write node js script to interact with the file system, and serve a web page from a File
27	Write node js script to build Your Own Node.js Module. Use require („http‟) module is a built-in Node module that invokes the functionality of the HTTP library to create a local server. Also use the export statement to make functions in your module available externally. Create a new text file to contain the functions in your module called, “modules.js” and add this function to return today‟s date and time.
28	Create a js file named main.js for event-driven application. There should be a main loop that listens for events, and then triggers a callback function when one of those events is detected.
29	Write node js application that transfer a file as an attachment on web and enables browser to prompt the user to download file using express js.
30	Case Studies on MEAN Stack Application Development

Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-562-MJ-TH	Major Elective	Web Services	Theory	2	2

Course Objectives:

- To understand the details of web services technologies like WSDL,UDDI, SOAP
- To learn how to implement and deploy web service client and server
- To explore interoperability between different frameworks
- To understand the concept of RESTful system

Course Outcomes:

On Completion of this course, student will be able to -

- CO1: Understand the web services and SOA
- CO2: Understand Web Services Architecture.
- CO3: Understand the working of SOAP and developing SOAP Web Services using Java.
- CO4: To get acquainted with the details of web services technologies like WSDL, UDDI.
- CO5: To understand the concept of RESTful services.

Course Contents:

Chapter-1	**Introduction to Web Services**	**Hours: 05**

1.1 Introduction
1.2 Need and definition of web services
1.3 Evolution and Emergence of Web Services
1.4 Basic operational model of web services
1.5 Tools and technologies enabling web services
1.6 The Service Oriented Architecture (SOA)
1.7 Use of web services in cloud
1.8 Benefits and challenges of using web services.

Chapter-2	**Web Services Architecture**	**Hours: 04**

2.1 Web services Architecture and its characteristics
2.2 Core building blocks of web services
2.3 Standards and technologies available for implementing web services

2.4 Basic steps of implementing web services.

Chapter-3	SOAP: Simple Object Access Protocol	Hours: 05

3.1 Inter-application communication and wire protocols
3.2 SOAP as a messaging protocol
3.3 Structure of a SOAP message with example
3.4 SOAP communication model
3.5 Building SOAP Web Services
3.6 Developing SOAP Web Services using Java
3.7 Error handling in SOAP
3.8 Advantages and disadvantages of SOAP.

Chapter-4	Describing, Registering and Discovering Web Services	Hours: 11

4.1 WSDL
- 4.1.1 WSDL in the world of Web Services
- 4.1.2 Anatomy of WSDL document
- 4.1.3 WSDL bindings, WSDL Tools
- 4.1.4 WSDL message exchange patterns
- 4.1.5 Limitations of WSDL.

4.2 UDDI
- 4.2.1 Service discovery
- 4.2.2 Role of service discovery in a SOA
- 4.2.3 Service discovery mechanisms
- 4.2.4 UDDI Registries
- 4.2.5 Uses of UDDI Registry
- 4.2.6 Programming with UDDI
- 4.2.7 UDDI data structures
- 4.2.8 Support for categorization in UDDI Registries
- 4.2.9 Enquiry API and Publishing API
- 4.2.10 Publishing information to a UDDI Registry
- 4.2.11 Searching information in a UDDI Registry
- 4.2.12 Deleting information in a UDDI Registry
- 4.2.13 Limitations of UDDI

Chapter-5	The REST Architectural Style	Hours: 05

5.1 Introducing HTTP
5.2 The core architectural elements of a RESTful system
5.3 Description and discovery of RESTful web services
5.4 Java tools and frameworks for building RESTful web services
5.5 JSON message format and tools and frameworks around JSON
5.6 Build RESTful web services with JAX-RS APIs
5.7 The Description and Discovery of RESTful Web Services

Reference Books:

1. Web Services & SOA Principles and Technology, Second Edition, Michael P. Papazoglou.
2. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India.
3. Developing Enterprise Web Services, S. Chatterjee, J. Webber, Pearson Education.
4. Gautam Shroff, "Enterprise Cloud Computing" ,Cambridge.
5. Building Web Services with Java, 2nd Edition, S. Graham and others, Pearson Edn., 2008.
6. Java Web Services, D.A. Chappell & T. Jewell, O"Reilly,SPD.
7. J2EE Web Services, Richard Monson-Haefel, Pearson Education.
8. Java Web Services Programming, R.Mogha, V.V.Preetham, Wiley India Pvt.Ltd.
9. XML, Web Services, and the Data Revolution, F.P.Coyle, Pearson Education.
10. Dr. Kumar Saurabh,"Cloud Computing", Wiley Publication
11. Borko Furht, "Handbook of Cloud Computing", Springer

Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-563-MJ-PR	Major Elective	Lab Course on CS-562-MJ-TH	Practical	2	4

Course Objectives:

- To understand the details of web services technologies like WSDL,UDDI, SOAP
- To learn how to implement and deploy web service client and server
- To explore interoperability between different frameworks
- To understand the concept of RESTful system

Course Outcomes:

On Completion of this course, student will be able to -
CO1: Understand the web services and SOA
CO2: Understand Web Services Architecture.
CO3: Understand the working of SOAP and developing SOAP Web Services using Java.
CO4: To get acquainted with the details of web services technologies like WSDL, UDDI.
CO5: To understand the concept of RESTful services.

Course Contents:

Assign No.	Name of Practical Assignment
1	Create 'Dynamic Web Project', which will host your web service functionality to find the factorial of given number and create 'Dynamic Web Project', which will host the client application that will send positive integer number and test the web service.
2	Create 'Dynamic Web Project', which will host your web service functionality to greet the user according to server time and create 'Dynamic Web Project', which will host the client application that will send user name and test the web service.
3	Create 'Dynamic Web Project', which will host your web service functionality to convert Celsius to Fahrenheit and create 'Dynamic Web Project', which will host the client application that will send Celsius and test the web service.
4	Create 'Dynamic Web Project', which will host your web service functionality for returning price of a stationary item and create 'Dynamic Web Project', which will host the client application that will send Name of any stationary item.

5	Create 'Dynamic Web Project', which will host your web service functionality to validate email id (use regular expression) and create 'Dynamic Web Project', which will host the client application that will send email id and test the web service.
6	Create 'Dynamic Web Project', which will host your web service functionality to validate user name and password and create 'Dynamic Web Project', which will host the client application that will send user name and password and test the web service
7	Create 'Dynamic Web Project', which will host your web service functionality to select staff details (use database for storing staff details (sno, sname, designation, salary)) and create 'Dynamic Web Project', which will host the client application that will send staff name and display the details.
8	Create 'Dynamic Web Project', which will host your web service functionality to return the percentage of a student when marks of five subjects are given as input and create 'Dynamic Web Project', which will host the client application that will send actor name and display the details.
9	Create 'Dynamic Web Project', which will host your web service functionality to validate mobile no (use regular expression: should contain only 10 numeric no) and create 'Dynamic Web Project', which will host the client application that will send mobile no and test the web service.
10	Create 'Dynamic Web Project', which will host your web service functionality to convert Rupees to Dollar, Pound, Euro,…..and create 'Dynamic Web Project', which will host the client application that will send amount in Rupees & type of conversion and tests the web service.
11	Create 'Dynamic Web Project', which will host your web service functionality to convert weight from kilograms to gram and create 'Dynamic Web Project', which will host the client application that tests the web service.
12	Create 'Dynamic Web Project', which will host your web service functionality to find area and volume of the rectangle and create 'Dynamic Web Project', which will host the client application that tests the web service.
13	Create 'Dynamic Web Project', which will host your web service functionality to find number of vowels in the given string and create 'Dynamic Web Project', which will host the client application that tests the web service.
14	Create 'Dynamic Web Project', which will host your web service functionality to convert decimal number to Binary, Octal,Hexa Decimal and create 'Dynamic Web Project', which will host the client application that will send decimal number & type of conversion and test the web service.
15	Create 'Dynamic Web Project', which will host your web service functionality to check whetherlogin success or fail (use database for storing username and password) and create 'Dynamic Web Project', which will host the client application that will send user name and password and test the web service.
Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**	

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-564-MJ-TH	Major Elective	ASP.NET Programming	Theory	2	2

Course Objectives:

- To understand the DOTNET framework
- Develop deep understanding of ASP.NET features
- Build strong concepts of OOP's and implement the same in ASP
- To understand the concept of multi-threading & files
- To understand and implement the controls & properties of Windows forms
- To Develop database centric applications

Course Outcomes:

On Completion of this course, student will be able to -

- CO1:Understand the features of Dot Net Framework along with the features of ASP
- CO2: Interpret and Develop Interfaces for real-time applications.
- CO3: Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in ASP programming language.
- CO4: Design & Implement the application using multithreading & File handling
- CO5: Design and Implement Windows Application using Windows Forms & tools application using Database in ASP
- CO6: Design and Implement Custom Application Using Windows Form & ADO.NET in ASP

Course Contents:

Chapter-1	**Introduction to ASP.NET**	**Hours: 02**
1.1 What is ASP.NET? 1.2 ASP.NET architecture and its components, 1.3 ASP.NET life cycle, 1.4 ASP.NET page life cycle, 1.5 Hello world Example in ASP.NET		
Chapter-2	**ASP.NET Sever controls**	**Hours: 07**

2.1 Types of server controls,
2.2 Working with button controls(image, link, radio button),
2.3 Text boxes, labels, literal, list controls(radio button list, checkbox list),
2.4 Panel, dropdown list, Data grid, Calendar, image map,
2.5 File upload,
2.6 Table,
2.7 Event handling in ASP.NET
2.8 Validation controls: Field validator, Compare validator, range validator, regular expression validator, custom validator,

Chapter-3	**Manage state in ASP.NET**	**Hours: 03**

3.1 View state,
3.2 Session state,
3.3 Application state,
3.4 Use of cookies and URL encoding

Chapter-4	**Web forms in ASP.NET**	**Hours: 03**

4.1 Creating a web page,
4.2 create and develop content page,
4.3 Access web page controls from content page

Chapter-5	**Database connection programming in ASP.NET**	**Hours: 07**

5.1 Fundamentals of database connectivity,
5.2 ADO.NET working,
5.3 Concurrency and the disconnected data architecture,
5.4 ASP.NET read database using SqlDataReader,
5.5 Functioning of insert, update, delete command in ASP.NET,
5.6 Connecting ASP.NET controls to data using DetailsView control,
5.7 FormView control, GridView control

Chapter-6	**Debugging and Error handling in ASP.NET page level**	**Hours: 03**

6.1 Debugging, tracing in ASP.NET,
6.2 Page level tracing, error handling,
6.3 ASP.NET unhandled exception,
6.4 ASP.NET error logging

Chapter-7	**Setup and deploy web applications of ASP.NET**	**Hours: 03**

7.1 Download and install IIS,
7.2 Deploy website in IIS,
7.3 Publishing ASP.NET website,
7.4 Unit testing

Chapter-8	**ASP.NET MVC**	**Hours: 02**

8.1 What is ASP.NET MVC?
8.2 Features of MVC, MVC architecture pattern,
8.3 Web form Vs MVC,
8.4 Advantages and disadvantages of ASP.NET MVC (model view control)

Reference Books:

1. Murach‟s ASP.NET 2.0 web programming by SPD publication
2. Profesional ASP.NET 2005/2008 by Wrox Publication

Examination Scheme: **IE : 15 Marks** **EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-565-MJ-PR	Major Elective	Lab course on CS-564-MJ	Prcatical	2	4

Course Objectives:

- To understand the DOTNET framework
- Develop deep understanding of ASP language features
- Build strong concepts of OOP's and implement the same in ASP.
- To understand the concept of multi-threading & files
- To understand and implement the controls & properties of Windows forms
- To Develop database centric applications using ADO.NET.

Course Outcomes:

On Completion of this course, student will be able to -

- CO1:Understand the features of Dot Net Framework along with the features of ASP
- CO2: Interpret and Develop Interfaces for real-time applications.
- CO3: Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in ASP programming language.
- CO4: Design & Implement the application using multithreading & File handling
- CO5: Design and Implement Windows Application using Windows Forms & tools application using Database in ASP
- CO6: Design and Implement Custom Application Using Windows Form & ADO.NET in ASP

Course Contents:

Assign No.	Name of Practical Assignment
1	Write an ASP.net program using Listview transfer item from on listview to another listview
2	Write an ASP.Net program to Validate student details form using validation control.
3	Write an ASP.net program on State management
4	Write web application in ASP.Net take two buttons on the page, a text box to enter string and a label to display the text stored from last session.

5	Create an ASP.Net application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts and shows to the user how many right answers were right and shows the result to user.
6	Write an ASP.net program, the user can enter 5 employee information in database and display in gridview
7	Write an ASP.Net program to Display Employee details (EmpID, Name, Designation, Joining Date, Mob.no, Gender) from database Edit, Delete information from GridView
8	Create an application of online test/quiz using MVC
9	Book Restaurant Table service using MVC
10	Design Crystal report on Employee‟s joining_date, Gender, designation.
Examination Scheme: **IE : 15 Marks**	**EE : 35 Marks**

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc. Computer Science**
Class	**F.Y.M.Sc (Computer Science)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
2	CS-581-OJT	On Job Training	On job Training (Internship)	Practical	4	-

Course Objectives:

- To provide students with practical, hands-on-experience in applying theoretical knowledge to real-world tasks
- To help students develop and enhance their skills, problem solving abilities and work culture of the industry
- To foster effective teamwork and collaboration skills
- To encourage students to build and expand their professional network by interactive with experienced experts and mentors in industry

Course Outcomes:

On Completion of this course, student will be able to -

- CO1: Enhance the knowledge related to various tools and technologies used in industry
- CO2: Improve the ability to solve complex problems independently and creatively
- CO3: Effectively utilize critical thinking and analytical skills in tackling real world challenges
- CO4: Effectively communicate and collaborate skills through interaction with team members and mentors.
- CO5: Get an experience in working on projects or related working within industry
- CO6: Develop the ability to document process, design, implementation and testing
- CO7: Familiar with specific industry domain relevant to internship
- CO8: Complete projects and tasks as per the predetermined objectives

Course Contents:

Sr. No.	Guidelines for On Job Training (OJT)
1	Student must start the OJT/Internship immediately after semester-II examination during the summer vacation
2	Student are expected to complete the IT related work/project within 120 hours assigned by organization (company/ industry/ consultancy/ institution)
3	The internship work may involve the IT related assignment(s) OR the maintenance of existing project OR the design/development of new project OR equivalent work

4	College should assign the mentors/guides for students to monitor the progress throughout the OJT
5	Students have to submit the weekly progress report duly signed by the concern authorities of organization to the assigned mentor
6	At the end of OJT, students should prepare the documentation and submit a report to the college in prescribed format
7	After completion, the final presentation and documentation will be evaluated by the examination panel as per the University norms
Examination Scheme: **IE : 30 Marks** **EE : 70 Marks**	

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-501-MJ-TH	MC	Database Systems and SQL	TH	04	04

Course Objectives:

- To be familiar with database management system
- To get acquainted with SQL and PL/SQL
- To understand advanced SQL features and procedural SQL
- To know the concept of triggers and assertions

Course Outcomes:

On completion of the course, student will be able to–

- Enumerate database applications
- Design E-R Model for given requirements and convert the same into database tables.
- Apply Normalization techniques for database design
- Formulate database queries using SQL
- Write Embedded and dynamic queries using SQL/PLSQL

Unit	Title and Contents	No. of lectures in Clock Hours
I	**Introduction of DBMS** ● DBMS Overview ● Advantages of DBMS ● Users of DBMS ● Applications of DBMS ● Data models - (Hierarchical, Network, ER, Relational), ● File system Vs. DBMS ● Data independence ● Levels of abstraction ● Architecture of DBMS ●DatabaseLanguages(DDL,DML,DCL)	10
II	**Conceptual Design (E-R model)** ● Overview of DB design ● Entity Types, Entity Sets, ● Attributes ,Attribute Types ● Relationship Types, Relationship Sets, Relationship Degree ● ER Diagrams, Naming Conventions(Attribute, Entity, Relationship), and Design Issues; ● ER-to-Relational Mapping, ● Schema Diagrams ● Characteristics of Specialization and Generalization ● keys, Constraints (Primary key, Foreign key, Check. Unique key, Not Null, Default etc)	14
III	**Relational Database Management Systems (RDBMS)** ● Introduction to Relational Database, Relational Database Design,DBMS vs RDBMS ● Functional Dependencies (Full functional dependency Partial functional dependency, Transitive functional dependency),Closure of set of Functional Dependency, Closure of set of attributes ● Decomposition,Properties of Relational Decomposition (Attribute Preservation, Dependency Preservation, Lossless join, No redundancy Non Additive Join Property.) ● Normalization, Need of Normalization, Normal form (1 NF,2NF,3NF,BCNF),● Case Studies	08
IV	**Introduction to SQL**	08

	●Introduction to SQL ● Data Types in SQL ● DDL commands (create, alter.drop,rename,desc) with examples ● DML command(insert,delete,update,select) ● DCL command(commit,rollback,grant,revoke) ● Basic structure of SQL SELECT query(Using BETWEEN, IN, OR,Like ,ORDER BY, GROUP BY and HAVING Clause,Distinct) ● Aggregate functions, ● Set operations	
V	**Intermediate SQL** ● Nested ,Sub-queries,(Using All,ANY), ● Joins and their type ● Grouping and summarizing information– A very common error with GROUP BY– The HAVING clause ● Writing queries on more than one table/multiple table -JOIN– Avoiding ambiguously named columns– Outer JOINs(LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN)– Using table aliases– SELF JOINS ● Overview of indexes,views,sequences ● Optimizing Queries with Indexes and views	10
VI	**PL/SQL, Embedded and Dynamic SQL** ● PL/PostgreSQL : Features,Advantages,Language structure,statements and Expressions ● Control flow, conditional statements, loops ● Cursors(Cursor attribute,Types-Implicit,explicit ,parameterized cursor,nesting of cursor) ● Stored procedure(creation,procedure call,implementation) ● Functions(creating ,calling function,passing parameters,returning a value) ● Handling errors and exceptions ● Triggers and Assertions	10

Reference Books:			
Sr. No	**Title of Books**	**Name of Author/s**	**Publisher**
1	Database System Concepts	Henry F. Korth, Abraham Silberschatz, S.Sudarshan	Tata McGraw-Hill Education 7th edition
2	Postgresql	Regina obe, Leo Hsu	OReilly publications3rdedition
3	Database Systems	Shamkant B. Navathe, RamezElmasri,	Pearson Higher Education
4	Database Management System	Raghu Ramakrishnan and Johannes Gehrke,	McGraw-Hill 3rd edition

Evaluation Scheme		
CIE	SEE	Total
30	70	100

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA 502 MJ-TH	MC	Python Programming and Data Structures	TH	04	04

Course Objectives:

- To introduce programming concepts using python
- Student should be able to develop Programming logic using python
- To develop basic concepts and terminology of python programming
- To test and execute python programs
- To be familiar with the concept of Data Structure.
- To learn the systematic way of solving problem
- To understand the different methods of organizing large amount of data
- To efficiently implement the different data structures
- To efficiently implement solutions for specific problems

Course Outcomes:

On completion of the course, student will be able to–

- Develop logic for problem solving
- Determine the methods to create and develop Python programs by utilizing the data
- structures like lists, dictionaries, tuples and sets.
- To be familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc.
- To write python programs and develop a small application project
- Design and implement Data structures and related algorithms

- Understand several ways of solving the same problem.
- To use well-organized data structures in solving various problems.
- To differentiate the usage of various structures in problem solution.
- Implementing algorithms to solve problems using appropriate data structures.

Unit	Title and Contents	No. of lectures in Clock Hours
I	**Basics of Python Programming** Introduction to python Features of Python, Identifiers, Reserved Keywords, Variables, Comments, Indentation in Python, Multiline Statements , Input, Output and Import Functions, Operators (Arithmetic, Comparison, Assignment, Bitwise, Logical, Membership, Identity), operator Precedence Data Types and Flow Control (Numbers, Strings, List, Tuple, Set, Dictionary , Data type conversion , decision Making (if, for, while, nested loops, control statements, types of loops)) Python tuples and sets Operations on tuples – Concept, operations and built-in unctions. Sets - Concept, operations and built-in functions. Python Dictionary(Concept (mutable),Creating and accessing values in a dictionary, Updating dictionary, delete dictionary elements, Properties of dictionary keys, built-in dictionary functions and methods	08
II	**Python Lists and Python Arrays** 2.1 Python Lists - concept, creating and accessing elements, updating & deleting lists, basic list operations, reverse, Indexing, slicing, built-in List functions, Functional programming tools - filter(), map(), and reduce(),,Using Lists as stacks and Queues, List comprehensions 2.2 Python Array - Concept of array-Array Representation, creating python array , accessing array elements.	06

	2.3 Types of Arrays – One , Two and Multidimensional array. 2.4 Array Operations-Traverse, Insertion, deletion, search and update 2.5 array slicing, python list vs array	
III	**Functions and Object oriented concepts** 3.1 Functions: Definitions and Uses, Function Calls, Parameters and Arguments, Variables and Parameters, Void Functions, Anonymous, Recursion, Lambda function Functional programming tools - filter(), map(), and reduce() 3.2 Python Classes / Objects Object oriented programming and classes in Python - creating classes, instance objects, accessing members ,Data hiding (the double underscore prefix) ,Built-in class attributes ,Recursive calls to methods ,Class variables, class methods, and static methods	06
IV	**Introduction to Data Structure, Sorting and Searching** 4.1 Concept , Need of Data Structure , Types of Data Structure 4.2. Algorithm analysis : definition, characteristics , Space complexity, time complexity 4.3 Asymptotic notation (Big O(Oh), Omega Ω) 4.4 Sorting algorithms with efficiency - Bubble sort, Insertion sort, Merge sort, Quick Sort 4.5 Searching techniques –Linear Search, Binary search	04
V	**Stacks and Queues** Stack : 5.1 Introduction 5.2 Representation- Using Arrays 5.3 Operations – init(), push(), pop(), isEmpty(), isFull(). 5.4 Application - infix to postfix, infix to prefix, postfix evaluation, 5.5 Simulating recursion using stack Queue : 5.6 Introduction 5.7 Representation - - Using Arrays	12

	5.8 Operations - init(), enqueue(), dequeue(), isEmpty(), isFull() 5.9 Types of Queue - Linear Queue, Circular Queue, Priority Queue, 5.10 Concept of doubly ended queue	
VI	**Linked List** 6.1 Introduction to Linked List 6.2 Implementation of Linked List – Static & Dynamic representation, 6.3 Types of Linked List – Singly, Doubly, Circular 6.4 Operations on Linked List - create, display, insert, delete, reverse, search, sort, concatenate & merge 6.5 Representing stacks and queues using linked lists	09
VII	**Trees** 7.1 Concept & Terminologies 7.2 Types - Binary tree, binary search tree , expression tree 7.3 Representation – Static and Dynamic 7.4 Operations on BST – create, Insert, delete, search , traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes , non recursive inorder traversal	09
VIII	**Graph** 8.1 Concept & terminologies 8.2 Graph Representation – Adjacency matrix, adjacency list, inverse Adjacency list, adjacency multi list, 8.3 Graph Traversals – Breadth First Search and Depth First Search	06

Reference Books:
1. An Introduction to Computer Science using Python 3 by Jason Montojo, Jennifer Campbell, Paul
Gries, The pragmatic bookshelf-2013
2. James Payne, “Beginning Python: Using Python and Python 3.1,Wrox Publication
3. Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication Learning
with Python “, Green Tea Press, 2002
4. Introduction to Problem Solving with Python by E balguruswamy,TMH publication2016
5.
Beginning Programming with Python for Dummies Paperback – 2015 by John Paul Mueller
5.Introducing Python- Modern Computing in Simple Packages – Bill Lubanovic, O,,Reilly Publication
6. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress

7. Data Structures – Horowitz, Sahani
8. Problem-Solving in Data Structures & Algorithms Using Python by Robert Karamagi
9. Algorithms & Data Structure in Python by Michael T. Goodrich, Roberto Tamassia, Michael H.
Goldwasser – Wiley Publication, student edition
10. Problem Solving in Data Structure & Algorithms using Python by Hemant Jain – Second Edition
Web references :
1. www.w3schools.com
2. www.tutorialspoint.com
3. www.javatpoint.com
4. www.geeksforgeeks.com
5. www.programiz.com
6. www.theserverside.com
7. www.educba.com
8. www.sanfoundry.com
9. www.prepbytes.com
10. www.codercampus.com

Evaluation Scheme		
CIE	SEE	Total
30	70	100

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-503-MJ-TH	MC	Operating Systems	TH	02	02

Course Objectives:

- To Understand the basic concepts of operating system.
- To study Architecture, File systems and basic operating system commands.
- To understand Processes, Threads and Deadlocks
- To analyze memory management schemes.
- To understand I/O management and File systems.

Course Outcomes:

On completion of the course, student will be able to–

- Explain basic concepts of operating system
- Describe algorithms for process, memory and disk scheduling
- Apply technique for inter-process communication and Multithreading.
- Implement concept of critical-section
- Compare and contrast deadlock avoidance and prevention.
- Use functions for file system management

Unit	Title and Contents	No. of lectures in Clock Hours
I	**Introduction**	04

	1.1 Introduction to Operating Systems, Different services provided by Operating System to Users. 1.2 Introduce the concept of Process, Process States, Process Control Block, User Interface, System Calls. 1.3 Introduction to Linux Operating System - Features of Linux, Architecture of the Linux, Introduction to File System and Process Environment.	
II	**File System** 2.1 File Concept, File Attribute, File Operations, File Types, File Structure 2.2 Access Methods - Sequential Access Method, Direct Access Method, Other Access Methods 2.3 Directory overview, Single level directory, Two level directory, Tree structure directory, Acyclic graph directory, General graph directory 2.4 File System Structure and Implementation - Partitions and Mounting, Virtual File Systems 2.5 Allocation Methods - Contiguous allocation, Linked allocation, Indexed allocation 2.6 Free Space Management – Bit vector, Linked list, Grouping, Counting, Space maps	06
III	**Process Scheduling and Multithreading** 3.1 Process Scheduling – Scheduling queues, Schedulers, context switch 3.2 Operations on Process – Process creation with program using fork(), Process termination 3.3 Interprocess Communication – Shared memory system, Message passing systems 3.4 Multithreaded Programming – Overview, Multithreading Models 3.5 Basic Concept – CPU-I/O burst cycle, CPU Scheduler, Pre-emptive Scheduling, Dispatcher 3.6 Scheduling Criteria 3.7 Scheduling Algorithms – FCFS, SJF, Priority scheduling, Round robin scheduling, Multiple queue scheduling, Multilevel feedback queue scheduling	06

IV	**Deadlock** 4.1 System Model 4.2 Deadlock Characterization – Necessary Conditions, Resource Allocation Graph 4.3 Deadlock Prevention 4.4 Deadlock Avoidance - Safe state, Resource-Allocation-Graph Algorithm, Banker's Algorithm 4.5 Deadlock Detection 4.6 Recovery from Deadlock – Process Termination, Resource Preemption	06
V	**Memory Management** 5.1 Introduction – Requirement of Memory management, Logical and Physical Address Space, Static and dynamic Loading, Static and Dynamic Linking 5.2 Memory Management Techniques- Contiguous memory management schemes, On-Contiguous memory management schemes 5.3 Swapping- Definition, Benefits of swapping 5.4 Memory allocation- Low Memory, High Memory 5.5 Partition Allocation- Best Fit, First Fit, Worst Fit, Next Fit 5.6 Paging- Use of Paging, 5.7 Fragmentation- External & Internal Fragmentation 5.8 Segmentation-Virtual Memory Segmentation, Simple Segmentation 5.9 Dynamic Loading, Dynamic Linking	08

Reference Books:
1. Operating Systems Achyut S. Godbole Tata McGraw Hill 2nd edition.
2. Operating Systems D.M. Dhamdhere Tata McGraw Hill 2nd edition.

NEP-CBCS-2024-25F.Y M.Sc. (Computer Applications)

15

3. Understanding Operating System: Flynn & Mctloes 4th edition, thomson.
4. Operating Systems Design & implementation Andrew S. Tanenbam, Albert S. Woodhull Pearson.
5. Operating System Concepts (7th Ed) by silberschatz and Galvin, Wiley, 2000.
6. Operating Systems (5th Ed) – Internals and Design Principles by William Stallings, Prentice Hall,
2000.

7. Operating System Concepts (2nd Ed) by James L. Peterson, Abraham Silberschatz, Addison –
Wesley.
8. Computer Organisation and Architecture (4th Ed) by William Stallings, Prentice Hall India, 1996.
9. Modern Operating Systems by Andrew S Tanenbaum, Prentice hall Inida, 1992.
10. UNIX – Sumitabha Das 11.Unix Shell Programming – Yashwant Kanetkar, BPB publications.

Evaluation Scheme		
CIE	SEE	Total
15	35	50

\

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-504-MJ-PR	MC	Lab course Based on CA 501 MJ-TH & CA 503 MJ-TH	PR	02	04

Course Objectives:

- To understand basic database management operations.
- To design E-R Model for given requirements and convert the same into database tables.
- To get acquainted with SQL and PL/SQL commands

Course Outcomes:

On completion of the course, student will be able to–
- Create database tables in postgreSQL.
- Write and execute simple, nested queries.

Assign No.	Practical Assignment using C Programming
1	To create simple tables with only the primary key constraint (as a table level constraint & as a field level constraint) (include all data types),
2	To create more than one table, with referential integrity constraint, PK constrain, Check constraint, Unique constraint , Not null constraint
3	To drop a table, alter schema of a table, insert / update / delete records using tables created in previous Assignments. (use simple forms of insert / update / delete statements)

4	To query the tables using simple form of select statement Select <field- list> from table [where <Condition> order by & <field list>] Select <field-list, aggregate functions> from table [where <condition> group by <> having order by <>]
5	To query table, using set operations (union, intersect)
6	To Write cursor and trigger, function and stored procedure
7	To implement scheduling algorithms like FCFS, RR, SJF
8	To implement bankers algorithm

Evaluation Scheme		
CIE	SEE	Total
15	35	50

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-505-MJ-PR	MC	Lab course based on CA 502 MJ-TH	PR	02	04

Assign No.	Practical Assignment using C Programming
1	BASIC PYTHON 1) Write a Python Program to Calculate the Average of Numbers in a Given List. 2) 2) Write a program which accepts 6 integer values and prints "DUPLICATES" if any of the values entered are duplicates otherwise it prints "ALL UNIQUE". Example: Let 5 integers are (32, 10, 45, 90, 45, 6) then output "DUPLICATES" to be printed. 3) Write a program to display following pattern. 1 2 3 4 5 6 7 8 9 10
2	PYTHON TUPLES 1. Reverse the following tuple aTup = (10, 20, 30, 40, 50) 2. Write a Python program to create a list of tuples with the first element as the number

	and second element as the square of the number. 3. Copy element 44 and 55 from the following tuple into a new tuple tuple1 = (11, 22, 33, 44,55, 66) 4. Write a Python program to get the 5th element from front and 5th element from last of a tuple. 5. Write a Python program to find the repeated items of a tuple. 6. Write a Python program to check whether an element exists within a tuple.
3	PYTHON SETS 1. What is the output of following program: sets = {1, 2, 3, 4, 4} print(sets) 2. Write a Python program to do iteration over sets. 3. Write a Python program to add and remove operation on set. 4. Write a Python program to find maximum and the minimum value in a set.
4	PYTHON DICTIONARY 1. Write a Python program to combine two dictionary adding values for common keys. Sample Dictionary: d1={'a':100,'b':200,'c':300} d2={'a':300,'b':200,'d':400} Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300}) 2. Write a Python script to generate and print a dictionary that contains a number (Between 1 and n) in the form (x, x*x). Sample Dictionary (n = 5) Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25} 3. Write a Python program to create a dictionary from a string. Sample- String:'W3resource' Expected output: {'3': 1, 's': 1, 'r': 2, 'u': 1, 'w': 1, 'c': 1, 'e': 2, 'o': 1}
5	PYTHON ARRAY 1. Write a python program to create an array of 5 integers and display the array elements. Access individual elements through indexes 2. write a python program to get the number of occurrences of specified elements in an array

	3. Write a python program to reverse the order of the items in the array
6	PYTHON FUNCTIONS 1. Write a python function to sum of all the elements in a list 2. Write a python function to calculate the factorial of a number.the function accept the number as an argument. 3. Write a python function to check whether a number falls within a given range. 4. Write a python function that takes a list and returns a new list with distict elements from the first list Sample list:[1, 2 , 2, 3, 3, 3, 3, 4, 5] Unique list:[1, 2, 3, 4, 5]
DATA STRUCTURE ASSIGNMENT	
7	Searching Algorithms - Implementation of searching algorithms to search an element using: Linear Search, Binary Search
8	Sorting Algorithms - Implementation of sorting algorithms: Bubble Sort, Insertion Sort , Quick Sort, Merge Sort
9	Singly Linked List -1. Dynamic implementation of Singly Linked List to perform following operations: Create, Insert, Delete, Display, Search, Reverse 2. Create a list in the sorted order.
10	Doubly Linked List - Dynamic implementation of Doubly circular Linked List to perform following operations: Create, Insert, Delete, Display, Search.
11	Linked List Applications - Merge two sorted lists.
12	Stack - Static and Dynamic implementation of Stack to perform following operations: Init, Push, Pop, Isempty, Isfull
13	Applications of Stack - 1. Implementation of an algorithm that reverses string of characters using stack and checks whether a string is a palindrome. 2. Infix to Postfix conversion. Evaluation of postfix expression
14	Linear Queue - Static and Dynamic implementation of linear Queue to perform following operations: Init, enqueue, dequeue, IsEmpty, IsFull.
15	Circular and Priority Queue 1. Implementation of circular queue 2. Implementation of priority queue
16	Tree Travarsals, operations etc
17	Calculate indegree and out degree of a given graph

Evaluation Scheme		
CIE	SEE	Total
15	35	50

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-510-MJ-TH	ME	Java Programming	TH	02	02

Course Objectives:

- To learn implementation of object-oriented concepts with Java.
- To understand collection classes and interfaces.
- To know the process of application development using Graphical User Interface (GUI)
-

Course Outcomes:

On completion of the course, student will be able to–

- Identify classes, objects, class members and relationships for a given problem.
- Design end to end applications using object-oriented constructs.
- Apply collection classes for storing java objects.
- Use Java APIs for program development.
- Handle abnormal termination of a program using exception handling

Unit	Title and Contents	No. of lectures in Clock Hours
I	Introduction of Java 1.1 A Short History of Java 1.2 Features of Java 1.3 Java Environment – Compiler, Interpreter, JVM 1.4 Structure of java program	04

	1.5 Data types, Variables, Operators, Keywords, Naming Convention 1.6 Decision Making (if, switch), Looping (for, while) 1.7 Type Casting 1.8 Array, Types of Arrays - One Dimensional arrays - Two-Dimensional array 1.9 Accepting input using Command line arguments 1.10 Accepting input from console (Using BufferedReader and Scanner	
II	2.1 Introduction to classes and objects 2.2 Defining Your Own Classes 2.3 Access Specifiers (public, protected, private, default) 2.4 Array of Objects 2.5 Constructor, types of constructor (default and parameterized) , Overloading Constructors and use of 'this' Keyword 2.6 static block, static fields and methods 2.7 Predefined class – Object class methods (equals (), toString(), hashcode(), getClass()) 2.8 Garbage Collection (finalize() Method)	04
III	**Inheritance, Interface and Package** Inheritance 3.1 Inheritance Basics (extends Keyword) and Types of Inheritance 3.2 Superclass, Subclass and use of super Keyword 3.3 Method Overriding and runtime polymorphism 3.4 Use of final keyword related to variable, method and class 3.5 Use of abstract class and abstract methods Interface 3.6 Defining and Implementing Interfaces 3.7 Runtime polymorphism using interface Packages 3.8 Creating, Accessing and using Packages	08
IV	**Collection, Exception Handling and I/O**	08

	Collections 4.1 Wrapper Classes 4.2 Introduction to the Collection framework 4.3 List – ArrayList, LinkedList and Vector 4.4 Set - HashSet, TreeSet, and LinkedHashSet 4.5 Map – HashTable ,HashMap, LinkedHashMap, TreeMap 4.6 Interfaces such as Iterators, ListIterators, Enumerations Exception Handling 4.7 Exception class, Checked and Unchecked exception 4.8 Catching exception and exception handling – try, catch, finally, throw and throws, multiple catch block 4.9 Creating user defined exception I/O 4.10 String class(basic methods), String Buffer class 4.11 File class 4.12 DataInputStream and DataOutputStream class	
V	**Swing** 5.1 What is Swing? 5.2 The MVC Architecture and Swing 5.3 Layout Manager and Layouts, The JComponent class 5.4 Components – JLabel, JButton, JText, JTextArea, JCheckBox, JRadioButton, JList, JComboBox, JMenu and JPopupMenu Class, JMenuItem 5.5 Dialogs (Message, confirmation, input), JFileChooser 5.6 Event Handling: Event sources, Listeners – ActionListener, ItemListener 5.7 Mouse and Keyboard Event Handling, Adapters – MouseAdapter, KeyAdapter	07
Reference Books: .1) Core Java Volume I - Fundamentals By Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978- 0-13-516630-7 2) The Complete Reference By Herbert Shildt, 11th Edition, McGraw Hill Education, ISBN 978-260-		

44023-2
3) Java Beginners Guide By Herbert Shildt, 8 th Edition, McGraw-Hill Education ISBN 978-1- 260-
44021-8
4) Core Java Volume II – Fundamentals By Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-
013-516631-4
5) Java 2 Programming Black Book By Steven Holzner, DreamTech Press, ISBN 978-93-5119- 953-4
E-books:
1) The Complete Reference By Herbert Shildt
https://gfgc.kar.nic.in/sirmv-science/GenericDocHandler/138-a2973dc6-c024-4d81-be6d-5c3344f232ce.pdf
2) Java 2 Programming Black Book By Steven Holzner
https://idoc.pub/documents/java-2-black-book-steven-holzner-vyly2rmq9v4m

Evaluation Scheme		
CIE	SEE	Total
15	35	50

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

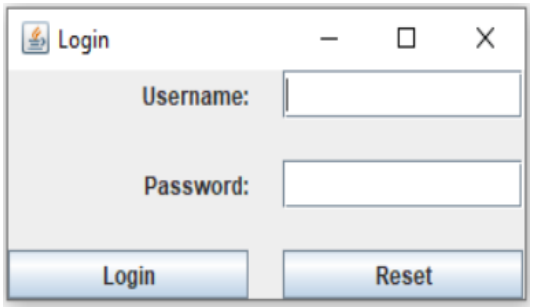
Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-511-MJ-PR	ME	Lab Course based on CA 510 MJ-TH	PR	02	04

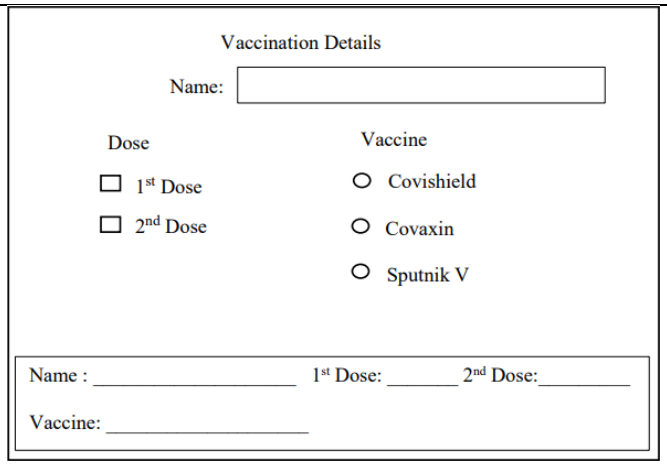
Assign No.	Practical Assignment using C Programming
1	Introduction To JAVA 1. Write a Java program to accept a number from user and generate multiplication table of a number. Accept number using Buffered Reader class. 2. Write a Java Program to Reverse a Number. Accept number using command line argument. 3. Write a Java program to print the sum of elements of the array. Also display array elements in ascending order. 4. Write a Java program to print the factors of a given number. (Use Scanner class). 5. Write a Java program to accept a number from user and print all prime numbers up to that number (Use Buffered Reader class). 6. Write a Java Program to Display Armstrong Numbers Between range. Accept range from user. 7. Write java program to check whether number is Perfect or not. 8. Write Java program to find multiplication of two matrix. Accept matrix from user.
2	Classes and Objects

	1. Define a class MyNumber having one private integer data member. Write a default constructor initialize it to 0 and another constructor to initialize it to a value. Write methods isNegative, isPositive, isOdd, iseven. Use command line argument to pass a value to the object and perform the above operations. 2. Write a program to create class Account (accno, accname, balance). Create an array of 'n' Account objects. Define static method “sortAccount” which sorts the array on the basis of balance. Display account details in sorted order. 3. Write a program which define class Product with data member as id, name and price. Store the information of 5 products and display the name of product having minimum price (Use array of object). 4. Write a program which define class Employee with data member as id, name and salary Store the information of 'n' employees and display the name of employee having maximum salary (Use array of object). 5. Define a class student having rollno, name and percentage. Define Default and parameterized constructor. Accept the 5 student details and display it. (Use this keyword). 6. Write a program create class as MyDate with dd,mm,yy as data members. Write parameterized constructor. Display the date in dd-mm-yy format. (Use this keyword). 7. Define a class Student with attributes rollno and name. Define default and parameterized constructor. Keep the count of Objects created. Create objects using parameterized constructor and display the object count after each object is created.
3	Inheritance, Interface and Package Inheritance 1. Define a “Point” class having members – x,y(coordinates). Define default constructor and parameterized constructors. Define two subclasses “ColorPoint” with member as color and subclass “Point3D” with member as z (coordinate). Write display method to display the details of different types of Points 2. Define a class Employee having members – id, name, salary. Define default constructor. Create a subclass called Manager with private member bonus. Define methods accept and display in both the classes. Create “n” objects of the Managerclass and display the details

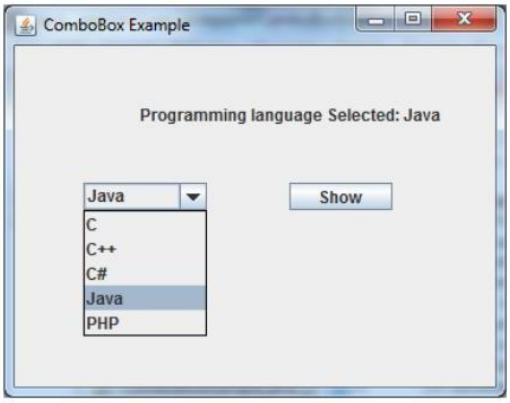
	of the worker having the maximum total salary (salary + bonus). 3. Write a Java program to create a super class Employee (members – name, salary). Derive a sub-class as Developer (member – projectname). Derive a sub-class Programmer (member – proglanguage) from Developer. Create object of Programmer and display the details of it. Implement this multilevel inheritance with appropriate constructor and methods. 4. Write a Java program to create a super class Vehicle having members Company and Price. Derive two different classes LightMotorVehicle (mileage) and HeavyMotorVehicle (capacity_in_tons). Accept the information for "n" vehicles and display the information in appropriate form. While taking data, ask user about the type of vehicle first 5. Define an abstract class Staff with members name and address. Define two sub- classes of this class – FullTimeStaff (members - department, salary, hra - 8% of salary, da – 5% of salary) and PartTimeStaff (members - number-of-hours, rate-per- hour). Define appropriate constructors. Write abstract method as calculateSalary() in Staff class. Implement this method in subclasses. Create n objects which could be of either FullTimeStaff or PartTimeStaff class by asking the user 's choice. Display details of all FullTimeStaff objects and all PartTimeStaff objects along with their salary.6. Create an abstract class Shape with methods area & volume. Derive a class Cylinder (radius, height). Calculate area and volume. Interface 1. Define an interface "Operation" which has methods area (), volume (). Define a constant PI having a value 3.142. Create a class circle (member – radius), cylinder (members – radius, height) which implements this interface. Calculate and display the area and volume. 2. Define an Interface Shape with abstract method area (). Write a java program to calculate an area of Circle and Sphere. (Use final keyword). Packages 1. Create a package named "Series" having three different classes to print series: a. Fibonacci series b. Cube of numbers c. Square of numbers Write a java program to

	generate “n” terms of the above series. Accept n from user. 2. Create a package “utility”. Define a class Capital String under “utility” package which will contain a method to return String with first letter capital. Create a Person class (members – name, city) outside the package. Display the person’s name with first letter as capital by making use of Capital String. 3. Write a package game which will have 2 classes Indoor & Outdoor. Use a function display () to generate the list of players for the specific game. Use default & parameterized constructor
4	Collection, Exception Handling and I/O Collections 1. Construct a linked List containing names of colours: red, blue, yellow and orange. Then extend the program to do the following: i. Display the contents of the List using an Iterator ii. Display the contents of the List in reverse order using a ListIterator iii. Create another list containing pink and green. Insert the elements of this list between blue and yellow 2. Write a program to accept ‘n’ integers from the user & store them in an Array List collection. Display the elements of Array List. 3. Accept 'n' integers from the user and store them in a collection. Display them in the sorted order. The collection should not accept duplicate elements. (Use a suitable collection). Search for a particular element using predefined search method in the Collection framework. 4. Create a Hash table containing Employee name and Salary. Display the details of the hash table. 5. Create a java application to store city names and their STD codes using an appropriate collection. i. Add a new city and its code (No duplicates) ii. Remove a city from the collection iii. Search for a cityname and display the code Exception Handling 1. Write a java program to accept a number from the user, if number is zero then throw user defined exception ―Number is 0, otherwise check whether no is prime or not. 2. Write a java program to accept Doctor Name from the user and check whether it is valid or not. (It should not contain digits and special symbol) If it is not valid then throw user

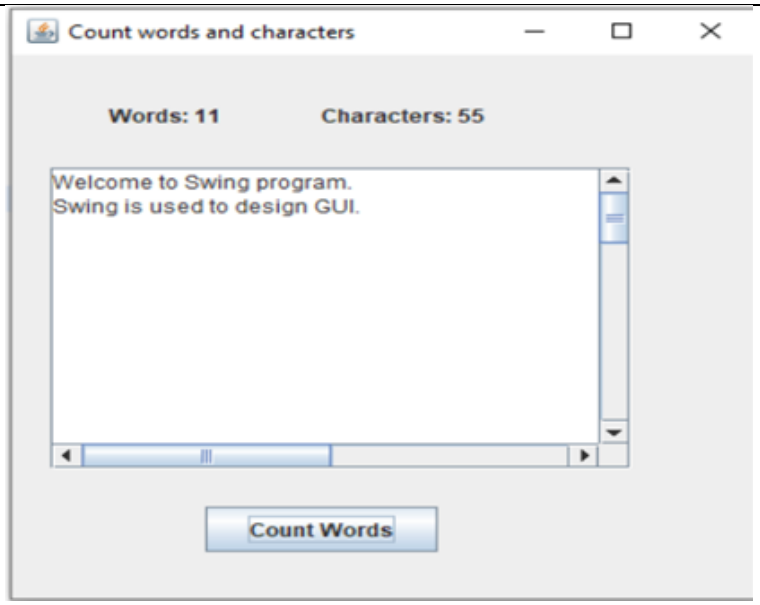
	defined Exception - Name is Invalid -- otherwise display it 3. Define a class MyDate (day, month, year) with methods to accept and display a MyDate object. Accept date as dd, mm, yyyy. Throw user defined exception "InvalidDateException" if the date is invalid. Examples of invalid dates : 12 15 2015, 31 6 1990, 29 2 2001. 4. Write a class Driver with attributeslicense_no, name, address and age. Initialize values through the parameterized constructor. If age of Driver is less than 18 then user-defined exception should be generated —Age is below 18 years – 5. Write a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception —Age Not Within The Range. If name contains numbers or special symbols raise exception —Name not valid I/O 1. Write a java program that displays the number of characters, lines and words of a file. 2. Write a java program to accept details of n customers (c_id, cname, address, mobile_no) from user and store it in a file (Use DataOutputStream class). Display the details of customers by reading it from file. (Use DataInputStream class). 3. Write a program to read the contents of "abc.txt" file. Display the contents of file in uppercase as output
5	**Swing** 1. Write a java program to design a following GUI. Use appropriate Layout and Components.  2. Write a java program to design a following GUI. Use appropriate Layout and Components



3. Write a java program to implement a simple arithmetic calculator. Perform appropriate validations
4. Write a Program to design following GUI by using swing component JComboBox. On click of show button display the selected language on JLabel.



5. Write a program to design following GUI using JTextArea. Write a code to display number of words and characters of text in JLabel. Use JScrollPane to get scrollbars for JTextArea.

Evaluation Scheme		
CIE	SEE	Total
15	35	50

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-512-MJ-TH	ME	Cloud Computing	TH	02	02

Course Objectives:

- To understand the principles and paradigm of Cloud Computing
- To appreciate the role of Virtualization Technologies
- Ability to design and deploy Cloud Infrastructure
- Understand Advanced Techniques and cloud security issues and solutions

Course Outcomes:

On completion of the course, student will be able to–

- Understand the different Cloud Computing environment
- Analyze virtualization technology and install virtualization software
- Develop and deploy applications on Cloud
- Use advance techniques and apply security in Cloud Computing

Unit	Title and Contents	No. of lectures in Clock Hours
I	**Introduction to Cloud Computing** Overview, Layers and Types of Cloud, Desired Features of a Cloud, Benefits and Disadvantages of Cloud Computing, Cloud Infrastructure Management, Infrastructure as a Service Providers,	08

	Platform as a Service Providers, Multitenant Technology. Cloud-Enabling Technology: Broadband Networks and Internet Architecture, Data Center Technology, Virtualization Technology. Cloud Deployment Models.	
II	**Virtualization** Introduction to Virtualization Technologies, Load Balancing and Virtualization, Understanding Hyper visors, Virtual Machines Provisioning and Manageability Virtual Machine Migration Services, Provisioning in the Cloud Context	06
III	**Programming, Environments and Applications** Features of Cloud and Grid Platforms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments, Applications: Moving application to cloud, Microsoft Cloud Services, Google Cloud Applications, Amazon Cloud Services, Cloud Applications.	08
IV	**Advanced Techniques and Security in The Cloud** Future Trends in cloud Computing, Mobile Cloud, Comet Cloud. Containers, Docker, and Kubernetes, Introduction to DevOps. Security Overview – Cloud Security Challenges and Risks – Software-as-a-Service Security – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security - Identity Management and Access Control, Disaster Recovery in Clouds.	08

Reference Books:

1. Brian J.S. Chee and Curtis Franklin, "Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center", CRC Press, ISBN:9781439806128
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, "Mastering Cloud Computing", McGraw Hill Education, ISBN-13:978-1-25-902995-0

3. Dr. Kris Jamsa, “Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more”, Wiley Publications, ISBN: 978-0-470-97389-9

4.https://sjceodisha.in/wp-content/uploads/2019/09/CLOUD-COMPUTING-Principles- andParadigms.pdf

5.https://arpitapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf

Cloud Computing https://onlinecourses.nptel.ac.in/noc21_cs14/preview?

Evaluation Scheme		
CIE	SEE	Total
15	35	50

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-513-MJ-PR	ME	Lab Course on CA-512-MJ-TH	PR	02	04

Course Objectives:

- To understand the principles and paradigm of Cloud Computing
- To appreciate the role of Virtualization Technologies
- Ability to design and deploy Cloud Infrastructure
- Understand Advanced Techniques and cloud security issues and solutions

Course Outcomes:

On completion of the course, student will be able to–
- Understand the different Cloud Computing environment
- Analyze virtualization technology and install virtualization software
- Develop and deploy applications on Cloud
- Use advance techniques and apply security in Cloud Computing

Assign No.	Practical Assignment
1.	Working and Implementation of Infrastructure as a service
2.	Working and Implementation of Software as a service
3.	Working and Implementation of Platform as a services

4.	Practical Implementation of Storage as a Service
5.	Installation and Configuration of Virtualization Using KVM
6.	Working of Google drive to make spreadsheet and notes.
7.	Write a program for web feed.
8.	Implementation of Virtualization in cloud computing to learn Virtualization Basics, Benefits of Virtualization in Cloud using Open Source Operating System.
9.	Execute the step to Demonstrate and implementation of cloud on single sign on.
10.	Installation and configuration of cloud Hadoop and demonstrate simple query
11.	Installing and Developing Application Using Google App Engine
12.	Case study on Amazon EC2/Microsoft Azure/Google Cloud Platform
13.	Design an Assignment based on working with Manjrasoft Aneka Software.
14.	Design and Develop Custom Application (Mini Project) using Salesforce Cloud

Evaluation Scheme		
CIE	SEE	Total
15	35	50

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science and Technology**
Program	**M.Sc CA**
Class	**F.Y. M.Sc. (Computer Applications)**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
I	CA-531-RM-TH	RM	Research Methodology	TH	04	04

Course Objectives:

- To investigate some existing situation or problems, explore and analyze it.
- To test hypothesis or theory.
- To identify patterns or trends related to the problem.
- To discover the truth and fact.
- To study the process of quantitative and qualitative data collection.

Course Outcomes:

On completion of the course, student will be able to–

- Understand and comprehend the basics in research methodology.
- Formulate research aims and objectives
- Organize and conduct research (advanced project) in a more appropriate manner.
- Develop and practice the skills necessary to conduct, review, and publish research.
- Write a research report and thesis.

Unit	Title and Contents	No. of lectures in Clock Hours
I	**Introduction to Research** • Definition of Research	03

	• Characteristics of Research • Objectives of Research • Nature of Research • Importance of Research • Relevance of Research • Restrictions in Research • Research Process • Difference between Research Method and Research Process	
II	**Scientific Method** • Introduction • Method to Eliminate Uncertainty • Scientific Method • Steps in Scientific Method • Distinction between Scientific Method & Non-Scientific Method • Difficulties encountered in Scientific Method Research • Inductive v/s Deductive Logic	08
III	**Types and Methods of Research** • Introduction • Pure and Applied Research • Exploratory or Formulative Research • Descriptive Research • Diagnostic Research • Evaluation Studies • Action Research • Experimental Research • Analytical Study or Statistical Method	10

	• Historical Research • Surveys • Case Study • Field Studies • Research ethics • Plagiarism Tools	
IV	• Purpose of Literature Review • Literature Resources • Internet and literature review • The Research Problem • The Importance of Formulating a Research Problem • Steps in Formulation of Research Problem • Formulation of Objectives • Establishing Operational Definitions	10
V	**Hypothesis and Sampling** • What is Hypothesis? • Nature & Characteristics of Hypothesis • Significance of Hypothesis • Types of Hypothesis • Sources of Hypothesis • Characteristics of Good Hypothesis • What is Sampling? • Aims of Sampling • Characteristics of Good Sample • Basis of Sampling	10

	• Advantages of Sampling • Limitations of Sampling • Sampling Techniques or Methods • Probability Sampling Methods • Non-Probability Sampling Methods • Sample Design and Choice of Sampling Technique	
VI	**Data Collection Techniques** • Introduction • Distinction between Primary Data and Secondary Data • Data Collection Procedure for Primary Data • Methods of Data Collection – Observation, Questionnaire, Interview, Focus group discussion	06
VII	**Quantitative and Qualitative Data Analysis** • What is Quantitative Data? • Types of Quantitative Data • Data Coding o Visual Aids for Quantitative Data Analysis-Tables, Bar Charts, Scatter graph, Line Graph etc. • Use of Statistics for Quantitative Data Analysis	10

	○ Measures of Central Tendency-Mean, Median, Mode ○ Measures of Distribution-Range, Fractiles, Standard Deviation ○ Finding Relationships in the data-Chi-Square, t-test, ANNOVA(f-test),Z-test • What is Qualitative Data Analysis? • Analyzing textual and non-textual qualitative data • Grounded Theory • Computer-aided qualitative Analysis • Quantitative and Qualitative Data Analysis Tools	
VIII	**Presentation of the Research** • Writing up the research • Paper presentation in Conference/Journal/Symposium etc • Poster presentation in exhibition • Software demonstration • Case Study -Preparation of Sample Research Paper	03
Reference Books: 1. Researching Information Systems and Computing by Briony J Oates, SAGE SOUTH ASIA EDITION		

2. The Research Methods Knowledge Base, by William M. K. Trochim, James P. Donnelly
3. Introducing Research Methodology: A Beginner's Guide to Doing a Research Project , by Uwe Flick

Evaluation Scheme		
CIE	SEE	Total
30	70	100

The Poona Gujarati Kelavani Mandal's
HARIBHAI V. DESAI COLLEGE
of Arts, Science & Commerce (Autonomous)
Affiliated to Savitribai Phule Pune University
(Linguistic Minority Institution) AICTE NO. : 1-44457797714
ID No.: PU / PN / ASC / 057/ (1984)
NAAC Grade B++ (2.86 CGPA) ■ AISHE CODE : C-41829
Principal:
Dr. Rajendra G. Gurao
M.Sc., Ph.D.
Email: principal@hvdesaicollege.edu.in

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	**CA-551-MJ-TH**	Major Core	Web Technologies	Theory	04	04

Course Objectives:

- *To understand and learn HTML and CSS*
- *To learn PHP programming and database connectivity*
- *To understand and learn AJAX and XML*

Course Outcomes:

- *On completion of the course, student will be able to–*
- *Develop web based application using suitable client side and server side web*
- *technologies.*
- *Build Dynamic web site using server side PHP Programming and Database connectivity.*
- *Build applications using AJAX and XML*

Course Contents:

Chapter-1	**Introduction to Web Technology, HTML and CSS**	**Hours: 06**

- Introduction to Web Technologies (Define terms : web page, web site, Web Browser, Web
- Server, URL, www)
- How does the Website Works?
- Software to create your website (Traditional method and best website builder)
- What makes a good website?
- Client-Server and its Communication
- Internet-Basic, Internet Protocols (HTTP,FTP,IP)
- Overview of popular frameworks (React, Angular, Vue.js)

- Introduction to cloud services like AWS, Azure, or Netlify.
- Introduction to HTML (different tags)
- Introduction to CSS, CSS types.

Chapter-2	Chapter Name: Introduction to PHP	Hours: 08

- Introduction to PHP
- PHP - Lexical structure, Language basics.
- Echo, Print Statement
- Variables, Data Types
- Operators
- Control Structures
- Strings

Chapter-3	Chapter Name: Function and Array in PHP	Hours: 08

- Defining and calling a function
- Default parameters, Variable parameters, Missing parameters
- Variable function, Anonymous function
- Arrow Functions: Comparison between traditional anonymous functions and arrow functions; implications on this
- Higher-Order Functions: Concept of functions that take other functions as parameters
- Indexed Vs Associative arrays, Identifying elements of an array
- Storing data in arrays, Multidimensional arrays
- Extracting multiple values, Traversing arrays, Sorting Using arrays

Chapter-4	Chapter Name: Object Oriented Programming	Hours: 06

- Classes
- Objects
- Introspection
- Serialization
- Inheritance
- Interfaces
- Encapsulation

Chapter-5	Chapter Name: Web Techniques	Hours: 08

- Variables
- Server information
- Processing forms
- Setting response headers
- Maintaining state
- TLS (Transport Layer Security)
- OAuth and OpenID Connect

Chapter-6	Chapter Name: Databases	Hours: 10

- Using PHP to access a database
- Relational databases and SQL

- PEAR DB basics
- Advanced database techniques
- Real-Time Applications with WebSockets : Develop a real-time web application (such as a chat app or collaborative tool) using WebSockets for live updates and interactions.

Chapter-7	Chapter Name: JavaScript	Hours: 06

- Concept of script, Types of Scripts : client side scripting language and server side
- scripting language, Introduction to JavaScript
- Data types, Variables, comments in JavaScript , operators, control structures.
- Functions
- Event Handling in Java Scripts (Event types, dialogue boxes)
- Concept of array, how to use it in JavaScript ,JavaScript array method, types of an
- Array
- Concept of String
- DOM concept in JavaScript - Methods of document object, How to access field value by
- document object.

Chapter-8	Chapter Name: XML and Ajax	Hours: 08

- What is XML?
- XML document Structure
- PHP and XML
- XML parser
- The document object model
- The simple XML extension
- Changing a value with simple XML
- Understanding java scripts for AJAX
- AJAX web application model
- AJAX –PHP framework
- Performing AJAX validation
- Handling XML data using PHP and AJAX
- Connecting database using PHP and AJAX

Reference Books:

1. Steven Holzner, ”HTML Black Book”, Dremtech press.
2. Web Technologies, Black Book, Dreamtech Press
3. Web Applications : Concepts and Real World Design, Knuckles, Wiley-India
4. Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel Pearson
5. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
6. Beginning PHP 5 , Wrox publication
7. PHP web sevices, Wrox publication
8. AJAX Black Book, Kogent solution
9. Mastering PHP , BPB Publication
10. PHP cookbook, O'Reilly publication
11. PHP for Beginners, SPD publication 8. Programming the World Wide Web , Robert W Sebesta(3rd Edition)

Evaluation Scheme:

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week	
II	CA-552-MJ-TH	Major Core	Introduction to Data Science	Theory	04	04	

Course Objectives:

- *Provide students with knowledge and skills for data-intensive problem solving and scientific discovery*
- *Be prepared with a varied range of expertise in different aspects of data science such as data collection, visualization, processing and modeling of large data sets.*
- *Acquire good understanding of both the theory and application of applied statistics and computer science based existing data science models to analyze huge data sets originating from diversified application areas.*
- *Be better trained professionals to cater the growing demand for data scientists in industry.*

Course Outcomes: *On completion of the course, student will be able to–*

- *Perform Exploratory Data Analysis*
- *Obtain, clean/process, and transform data.*
- *Detects and diagnoses common data issues, such as missing values, special values,outliers, inconsistencies, and localization.*
- *Demonstrate proficiency with statistical analysis of data.*
- *Present results using data visualization techniques.*
- *Prepare data for use with a variety of statistical methods and models and recognize how the quality of the data and the means of data collection may affect conclusions.*

Course Contents:

Chapter-1	**Chapter Name: Introduction to Data Science**	**Hours: 10**

- Introduction to data science, The 3 V's: Volume, Velocity, Variety,

- Applications of Data Science
- The Data Science Lifecycle
- Essential Tools used in data science projects
- Types of Data
    - Structured, semi-structured, Unstructured Data,
    - Problems with unstructured data
- Data sources: Open Data, Social Media Data, Multimodal Data, Standard datasets
- Data Formats
    - Integers, Floats, Text Data, Text Files, Dense Numerical Arrays, Compressed or Archived Data, CSV Files, JSON Files, XML Files, HTML Files , Tar Files, GZip Files, Zip Files,
- Image Files: Rasterized, Vectorized, and/or Compressed

Chapter-2	**Chapter Name: Statistical Data Analysis**	**Hours: 18**

- Role of statistics in data science
- Descriptive statistics
    - Measuring the Frequency
    - Measuring the Central Tendency: Mean, Median, and Mode
    - Measuring the Dispersion: Range, Standard deviation, Variance, InterquartileRange
- Inferential statistics
    - Hypothesis testing, Multiple hypothesis testing, Parameter Estimation methods
- Measuring Data Similarity and Dissimilarity
    - Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal
    - Attributes, Proximity Measures for Binary Attributes, Dissimilarity of
    - NumericData: Euclidean, Manhattan, and Minkowski distances,
    - Proximity Measures for Ordinal Attributes
- Concept of Outlier, types of outliers, outlier detection methods: Z-Score, Local Outlier Factor (LOF), Isolation Forest, DBSCAN

Chapter-3	**Chapter Name: Data Preprocessing**	**Hours: 16**

- Data Objects and Attribute Types: What Is an Attribute?, Nominal , Binary, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes
- Data Quality: Why Preprocess the Data?Data munging/wrangling operations
- Cleaning Data - Missing Values, Noisy Data (Duplicate Entries, Multiple Entries for a Single Entity, Missing Entries, NULLs, Huge Outliers, Out-of- Date Data, Artificial Entries, Irregular Spacings, Formatting Issues - Irregular between Different Tables/Columns, Extra Whitespace, Irregular Capitalization,Inconsistent Delimiters, Irregular NULL Format, Invalid Characters, Incompatible Datetimes)
- Data Transformation – Rescaling, Normalizing, Binarizing, Standardizing,Label and OneHot Encoding
- Data reduction
- Data discretization

Chapter-4	**Chapter Name:Data Visualization**	**Hours: 16**

- Introduction to Exploratory Data Analysis
- Data visualization and visual encoding
- Data visualization libraries
- Basic data visualization tools
    - Histograms, Bar charts/graphs, Scatter plots, Line charts, Area plots, Pie charts,Donut charts, Pair plot
- Specialized data visualization tools

○ Box Plots, Bubble plots, Heat map, Dendrogram, Venn diagram, Treemap, 3d Scatter plots ○ Advanced data visualization tools Word cloud Visualization of geospatial data ○ Data Visualization types	
Reference Books: ● Data Science Fundamentals and Practical Approaches, Gypsy Nandi, Rupam Sharma, BPB Publications, 2020. ● The Data Science Handbook, Field Cady, John Wiley & Sons, Inc, 2017 ● Data Mining Concepts and Techniques, Third Edition, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, 2012. ● A Hands-On Introduction to Data Science, Chirag Shah, University of Washington Cambridge University Press ● https://dataheroes.ai/	
Evaluation Scheme **CIE : 30 Marks** **SEE : 70 Marks**	

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-553-MJ-TH	Major Core	Computer Networks	Theory	02	02

Course Objectives:

- *To understand the fundamental concepts of networking standards, protocols and technologies.*
- *To study different techniques for framing, error control, flow control and routing.*
- *To learn the role of protocols at various layers in the protocol stacks.*
- *To develop an understanding of modern network architectures from a design and performance perspective*

Course Outcomes: After successful completion of this course, learner will be able to-

- *Analyze the requirements for a given organization and select appropriate network architecture, topologies, transmission mediums and technologies.*
- *Analyze data flow between TCP/IP model using Application, Transport and Network Layer Protocols.*
- *Illustrate applications of Computer Network.*
- *Compare and contrast different routing and switching algorithms.*

Course Contents:

Chapter-1	**Chapter Name: Introduction to Data Communications Computer Networks**	**Hours: 06**

- Data communications,
  - Characteristics of Data Communication
  - Components of Data communication
- Data Representation – Text, Numbers, Images, Audio, Video
- Types of Data flow – Simplex, Half Duplex, Full Duplex

- Computer Networks applications –Business Application, Home Application, Mobile User
- Broadcast and point-to-point networks
- Network Topologies - Bus, Star, Ring, Mesh
- Network Types- LAN, MAN, WAN, internetworks
- Protocols and standards – Definition of a Protocol, Protocol standards: De facto and De jure
- OSI Model – layered architecture, peer-to-peer processes
- TCP/IP Model – layers and Protocol Suite
- Addressing - Physical, Logical, Port addresses, Specific addresses

Chapter-2	**Chapter Name:Physical Layer**	**Hours: 06**

- Analog and Digital data, Analog and Digital signals, Digital Signals-Bit rate, Bit length
- Baseband Transmission, Broadband Transmission
- Transmission Impairments– Attenuation, Distortion and Noise
- Data Rate Limits– Noiseless channel: Nyquist's bit rate, noisy channel : Shannon's Law
- Performance of the Network Bandwidth, Throughput, Latency (Delay), Bandwidth – Delay Product, Jitters
- Line Coding Characteristics, Line Coding Schemes–Unipolar -NRZ, Polar-NRZ-I, NRZ-L, RZ, Problems
- Transmission Modes, Parallel Transmission and Serial Transmission– Asynchronous and Synchronous
- Multiplexing- FDM and TDM
- Switching-Circuit Switching, Message Switching

Chapter-3	**Chapter Name:Data Link Layer**	**Hours: 05**

- Framing – Concept, Methods – Character Count, Flag bytes with Byte Stuffing,Starting & ending Flags with Bit Stuffing
- Error detection code – Hamming Distance, CRC
- Elementary data link protocols - Simplex stop & wait protocol, Simplex protocol for noisy channel,
- Sliding Window Protocols – 1-bit sliding window protocols,
- Pipelining – Go-Back N and Selective Repeat
- Random Access Protocols - ALOHA– pure and slotted, CSMA-1- persistent, p-persistent and nonpersistent CSMA/CD,CSMA/CA
- Controlled Access - Reservation, Polling and Token Passing

Chapter-4	**Chapter Name:**Network Layer	**Hours: 05**

- IPv4 addresses: Address space, Notation, Classful addressing, Classless addressing,
- IPv4: Datagram, Fragmentation, checksum, options
- IPv6 addresses: Structure, address space
- IPv6:packet format, Extension headers

Chapter-5	**Chapter Name:**Transport and Application Layer	**Hours: 08**

- Process-to-Process Delivery, Multiplexing and Demultiplexing
- User Datagram Protocol (UDP) - Datagram Format, Checksum, UDP operations, Use of
- UDP 6.3. Transmission Control Protocol (TCP) - TCP Services – Process to-Process
- Communication, Stream Delivery Service, Sending and Receiving Buffers, Segments, Full – Duplex Communication, Connection oriented service, Reliable service
- TCP Features – Numbering System, Byte Number, Sequence Number, Acknowledgement Number, Flow Control, Error Control, Congestion Control
- TCP Segment Format
- TCP Vs UDP
- Domain Name System (DNS) - Distribution of Name Space, DNS in the Internet

- E-MAIL - Architecture, User Agent, Message Transfer Agent - SMTP, Web Based Mail
- HTTP - HTTP Transaction

Reference Books:

1. Data Communications and Networking by Behrouz Forouzan, Fifth Edition, ISBN 978-0-07-337622-6 McGraw Hill.
2. Computer Networks, ANDREW S. Tanenbaum, Fifth Edition, ISBN-13: 978-0-13- 212695-3, Pearson

Evaluation Scheme

CIE : 15 Marks

SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-554-MJ-PR	Major Core	Lab Course based on CA-551-MJ-TH	Practical	02	04

Course Contents:

Practical Assignment : Set I (HTML and CSS)

- Write an HTML program to display the message "Exploring the Future of Web Development".
- Write an HTML program to display the word "Cascading Style Sheets" (CSS) in sizes h1 to h6, with each heading having a different background color.
- Write an HTML script to display the phrase "Learning Web Development" with different styles: bold for "Learning", italic for "Web", and underline with a strike for "Development". Apply separate effects on different words.
- Write an HTML script that uses an image of your favorite vacation destination as a background and displays a welcoming message on top of it.
- Create an HTML page with the following specifications: a. Title should be about "My Favorite Vacation Spot". b. Place the name of the vacation spot at the top of the page in large text and in blue color. c. Add names and images (as links) of activities available at your vacation spot, each styled in different colors and fonts. d. After clicking on the images, it should display fun facts about each activity.
- Write HTML code to display the following output:
    - Fruits
    - Apples
    - Bananas
    - Vegetables

- Carrots
- Spinach

- Write HTML code to display a list of different student clubs available at your college using both ordered and unordered lists, and categorize them by type (e.g., Academic, Cultural, Sports).
- Design a table that shows the monthly schedule of events for a specific club or organization within your college, including details such as event name, date, time, and location.
- Divide a screen in four equal part . Each frame shows : list of different activities conducted by your department.
- Design a admission form. which should contains : text box, multiline text box, a table which shows your academic record, radio button, check box, submit button etc.
- Write inline CSS program to display with background color pink with red colored text.
- Write internal CSS program to display with background color black with white colored text.
- Write external CSS program to display with background color sky blue with blue colored text.
- Write CSS using HTML which uses of text decoration, border, padding and margin.
- Write CSS using HTML which displays following output:

Positioning in CSS : Static, Relative, Fixed and Absolute

This div element has position: static;

This div element has position: relative;

- Write CSS using HTML which displays following output
- List Property in CSS

Unordered lists
o Coffee
o Tea
o Milk
 Apple
 Mango
 Banana
 Watermelon
Ordered list
I. Rose
II. Jasmine
III. Marigold
a. Sunflower
b. Tulip
c. Lily
d. Tuberose

- Write CSS using HTML which displays following output

Add a border to a table:

Firstname	Lastname
Ram	Joshi
Sham	Kulkarni

- Write CSS using HTML which displays following output : use image property

- Write CSS using HTML which displays following output

The display Property

Display : none

Display : inline

Web Technology !

Display : block

Web Technology !

Display : inline-block

Web Technology !

- Write CSS using HTML which displays following output : Use of Id and classes in CSS

Web Technology !

This paragraph is not affected by the style.

Web Technology !

This paragraph is affected by the style.

- Write CSS using HTML which displays following output : Use of and in CSS

The < span > element ! : I have a Red rose and dark Chocolate.

The < div >

Web Technology !

MCA (Science)

Computer Application !

Practical Assignment : Set II (Introduction to PHP)

- Write a PHP script for the following: a) Design a form to accept the details of 5 different items such as Item code, Item Name, Quantity, Sold, and Rate. b) Display a consolidated bill in tabular format that includes total amount for each item and overall total. Ensure that the form accepts items as a single string (comma-separated values) and utilize explode to process the data.
- Design an HTML form to accept a paragraph of text. Write a PHP script for the following: a) Write a function to calculate the total number of vowels in the input paragraph. b) Display a breakdown showing the occurrences of each vowel (A, E, I, O, U) from the input.

- Write a PHP script for the following. Design a form to accept a string and check whether the given string is a palindrome. Additionally, provide a feature that ignores spaces and is case-insensitive during the palindrome check.
- Write a PHP Script to accept a customer's full name from the user and do the following: a) Transform the customer's full name into all upper case letters. b) Capitalize the first letter of each word in the customer's name.
- Write a PHP script to generate and print Floyd's triangle, but modify it to allow for variable rows based on user input. Allow the user to specify how many rows of Floyd's triangle they would like to see.
- Write a PHP script that allows a user to input a URL. Upon submission, display the source code of the webpage at the specified URL.
- Write a PHP script that accepts a number from a user and uses a ternary operator to determine if the number is greater than 50, 30, or 20. Provide appropriate output messages based on the user's input.
- Write a PHP script to display a multiplication table for a user-input number. Design an HTML page that accepts a number and allows the user to specify how many rows of the multiplication table to display.
- Write a PHP script that accepts a number (up to one million) and converts it into words. Design an HTML page to accept the number.
- Write a PHP script to accept the details of an Employee (Name, Salary, Designation, Address, and join date) and display it on a new page in a well-formatted layout. Include validation to ensure salary is a positive number and the join date is in the correct format.

Practical Assignment : Set III (Function and Array)

- Write a PHP script to accept the number from user and Write a PHP function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.
- Design a HTML form to accept a string. Write a PHP function that checks whether a passed string is a palindrome or not?
- Design a HTML form to accept a string. Write a PHP script for the following. a)Write a function to count the total number of Vowels from the script. b) Show the occurrences of each Vowel from the script.
- Write a PHP script for the following: a) Design a form to accept two numbers from the users. b) Give option to choose an arithmetic operation (use Radio Button). c) Display the result on next form. d) Use concept of default parameter.
- Write a PHP script for the following: Design a form to accept two strings. Compare the two strings using both methods (= = operator & strcmp function). Append second string to the first string. Accept the position from the user; from where the characters from the first string are reversed. (Use radio buttons)
- Write a menu driven program to perform the following operations on an associative array: a) Display the elements of an array along with the keys. b) Display the size of an array

- Write a menu driven program the following operation on an associative array a) Reverse the order of each element's key-value pair. [Hint: array_flip()] b) Traverse the element in an array in random order. [Hint: shuffle()]
- Declare array. Reverse the order of elements, making the first element last and last element first and similarly rearranging other array elements.[Hint : array_reverse()]
- Write a menu driven program to perform the following stack related operations. a) Insert an element in stack. b) Delete an element from stack.[Hint: array_push(), array_pop()]
- Write a menu driven program to perform the following operations on associative arrays: a) Merge the given arrays. b) Find the intersection of two arrays. c) Find the union of two arrays. d) Find set difference of two arrays.
- Write a menu driven program to perform the following queue related operations a) Insert an element in queue b) Delete an element from queue c) Display the contents of queue

Practical Assignment : Set IV(Class and Object)

- Write a PHP program to define Interface shape which has two method as area() and volume (). Define a constant PI. Create a class Cylinder implement this interface and calculate area and Volume.
- Write a PHP script to create a Class shape and its subclass triangle, square and display area of the selected shape.(use the concept of Inheritance) Display menu (use radio button)

a) Triangle
b) Square
c) Rectangle
d) Circle

- Write PHP script to demonstrate the concept of introspection for examining object.
- Create a class named DISTANCE with feet and inches as data members. The class has the following member functions: convert_feet_to_inch() , convert_inch_to_feet() . Display options using radio button and display conversion on next page.
- Write a PHP program to create a class temperature which contains data members as Celsius and Fahrenheit . Create and Initialize all values of temperature object by using parameterized constructor . Convert Celsius to Fahrenheit and Convert Fahrenheit to Celsius using member functions. Display conversion on next page.
- Write a Calculator class that can accept two values, then add them, subtract them, multiply them together, or divide them on request.
- Write a PHP Script to create a super class Vehicle having members Company and price. erive 2 different classes LightMotorVehicle (members – mileage) and HeavyMotorVehicle (members – capacity-in-tons). Define 5 Object of each subclass and display details in table format.

Practical Assignment : Set V (Web Techniques)

- Write a PHP script to display following information using super global variable. a) Client IP Address. b) Browser detection/information. C) To check whether the page is called from 'https' or 'http'.
- Write a PHP script to keep track of number of times the web page has been access. [Use Session]

- Write a PHP script to accept username and password. If in the first three chances, username and password entered is correct then display second form with 'welcome message' otherwise display error message. [Use Session]
- Write a PHP script to accept Employee details (eno, ename, address) on first page. On second page accept earning (Basic, Da, HRA). On third page print Employee information(eno,ename, Address, BASIC, DA, HRA, TOTAL) [Hint: Use Session]
- Write a PHP script to check how many times the web page access.[Use cookies]
- Write a PHP script to change the preference of your web page like font style, font, size, font color, background color using cookie. Display selected settings on next page and actual implementation (with new settings) on third page.

Practical Assignment : Set-VI (Databases (MySQL))

- Consider the following entities and their relationship.

Doctor (doc_no, dname, address, city, area)
Hospital (hosp_no, hname, hcity)
Doctor-Hospital related with many-one relationship.
Create a RDB in 3NF for above and solve the following.
Using above database write a script in PHP to print the Doctor visiting to the hospital in tabular format. Accept hospital name from user[Use MySQL]

- Consider the following entities and their relationship. Student (stud_id, name, class) Competition(c_no,c_name, type) a) Relationship between student and competition In many-many with attributes rank and year. b) Create a RDB in 3NF for above and solve the following. c) Using above database write a script in PHP to accept a competition from user and display information of student who has secured 1st rank in that competition.
- Consider the following entities and their relationship Emp(e_no, ename, address, phone, salary) Dept(d_no, dname, location) Emp-Dept related with many-one relationship. Create a RDB in 3NF for above and solve the following. Using above database write a script in PHP which will a) Insert Employee records and Department records into respective tables. b) Print a salary statement in the format given below, for a given Department. [Hint : create a HTML form to accept Department name form user Maximum Salary Minimum Salary Sum salary.

Practical Assignment : Set VII (JavaScript)

- Write the JavaScript to convert temperature from Kelvin to Celsius and Fahrenheit.
- Modify the program to accept any number of subjects and calculate the average score along with the percentage.
- Write JavaScript to calculate compound interest over a specified number of years.
- Write the JavaScript to swap two values using an array. For example: if A = 100 and B = 200, demonstrate how to swap without using a temporary variable.
- Write the JavaScript to take input as a student's age and check eligibility for voting and driving in different regions (age requirements may vary).
- Write the JavaScript to determine how many leap years exist between two given years.
- Write the JavaScript to print the grade of a student based on different grading scales (letter grades, percentage brackets) using If-Else and switch statements.

- Modify the program to categorize students into different classes (first, second, third) based on the overall percentage.
- Write the JavaScript to accept a string representing a weekday and return whether that day is a weekday or weekend.
- Write the JavaScript to print the multiplication tables for numbers between 1 and n in a tabular format.

Practical Assignment : Set-VIII (XML and AJAX)

- Write a script to create XML file as 'Employee.xml'. The element of this xml file are as follows:

<Empdetails>
<Employee EMPno= Empname=>
<Salary>--------</Salary>
<Designation>-------</Designation>
</Employee>
</Empdetails>

- Write a PHP script to generate an XML in the following format in php.

<? Xml version='1.0'encoding=''ISO-8859-1''?>
<Book Store>
<Books>
<PHP>
<Title> Programming in PHP </ Title>
<Publication>O'RELLY<Publication>
</PHP>
<PHP>
<Title> Beginners PHP</ Title>
<Publication> WORX</Publication>
</PHP></Books>
</Book Store>

- Write a script to create XML file 'University.xml'. The element details of
- 'University.xml' Are as follows:

<Univ>
<Uname>----------</Uname>
<CITY>----------</CITY>
<Rank>----------<</Rank>
</Univ>

a) Store the details of at least 3 universities.
b) Link the 'University.xml' file to CSS and get well formatted output as given below.
i) Uname :
Color : black;
Font-family: copperplate G0thic Light;
Font size: 16pt;
Font:Bold;
ii) City and Rank
Color: Yellow;
Font-family: Arial;
Font-size : 12pt;

Font: Bold;

- Write a PHP Script to read 'BOOK.xml' file and print specific content of a file using DOMDocument parser. 'Book.xml' file should contain following information with at least 5 records with values. BookInfo : Book NO, Book Name, Author Name, Price, Year. [Note: Examiners can change the Book info file to Student info, Teacher info]
- Write a AJAX program to read contact. Dat file and print the contain of a file in a Tabular form when the user clicks on print button. Contact.dat file contain srno, name, residence number, mobile number, context/ relation. [Enter at least 3 record in contact.dat file] [Note: Examiner may change the contact. dat, dept.dat and provide proper structure of the file]
- Write AJAX program to print movie by selecting an actor's name. create table
  Movie and Actor with 1:M cardinality as follows:
  Movie (mno, mname, release_year) Actor(ano, aname) [USE MySQL]
- Write a AJAX program to search Student name according to the character typed and display list using array
- Write a AJAX program to print Teacher information from MySQL table
- Teacher: Teacher (Tno, Name, Subject, Research area). [Note: Examiner can change MySQL table]

Evaluation Scheme:
CIE : 15 Marks
SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-555-MJ-PR	Major Core	Lab Course based on CA-552-MJ-TH	Practical	02	04

1) Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.
2) Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.
3) Write a R program to create a simple bar plot of five subjects marks.
4) Write a R program to get the unique elements of a given string and unique numbers of vector.
5) Write a R program to multiply two vectors of integers type and length 3.
6) Write a R program to list containing a vector, a matrix and a list and give names to the elements in the list.
7) Write a R program to create a list containing a vector, a matrix and a list and give names to the elements in the list. Access the first and second element of the list.
8) Write a R program to create a list containing a vector, a matrix and a list and remove the second element.
9) Write a R program to merge two given lists into one list.
10) Write a R program to assign new names "a", "b" and "c" to the elements of a given list.
11) Write a R program to create an empty data frame.
12) Write a R program to create a data frame from four given vectors.
13) Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame.
14) Write a R program to save the information of a data frame in a file and display the information of the file.
15) Write a R program to create an ordered factor from data consisting of the names of months.
16) Write R program to find whether given number is positive or negative.
17) Write R program to read number and print corresponding day name in a week

18) Create a Matrix using R and Perform the operations addition, subtraction, multiplication.
19) Using R import the data from Excel/.CSV file and find mean, median, mode, quartiles.
20) Using R import the data from Excel/.CSV file and find standard deviation, variance and co-variance.
21) Write a R program to count the number of NA values in a data frame column.
22) Write a R program to call the (built-in) dataset air quality. Remove the variables 'Solar.R' and 'Wind' and display the data frame.
23) Write a R program to compare two data frames to find the row(s) in first data frame that are not present in second data frame
24) Write a R program to create a factor corresponding to height of women data set, which contains height and weights for a sample of women.
25) Write a R program to find nth highest value in a given vector.
26) Write an R program to sort a Vector in ascending and descending order.
27) Write an R program to extract first 10 English letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.
28) Write an R Program to calculate Decimal into binary of a given number.
29) Write an R program to convert a given matrix to a list and print list in ascending order.
30) Write an R program to create Data frames which contain details of 5 employees and display the details in ascending order.
31) Consider the inbuilt iris dataset i) Create a variable "y" and attach to it the output attribute of the "iris"dataset .ii) Create a barplot to break down your output attribute. iii) Create a density plot matrix for each attribute by class value.
32) Consider Weather dataset i) Selecting using the column number ii)Selecting using the column name iii) Make a scatter plot to compare Wind speed and temperature.
33) Write a script in R to create a list of students and perform the following
i) Give names to the students in the list. ii) Add a student at the end of the list.
iii) Remove the first Student.iv) Update the second last student

Evaluation Scheme
CIE : 15 Marks
SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-560-MJ-TH	Major Elective	Advanced Java Programming	Theory	02	02

Prerequisites:

- Basic knowledge of Java programming
- Understanding of Object-Oriented Programming concepts

Course Objectives:

- To learn database programming using Java
- To study web development concept using Servlet and JSP
- To develop a game application using multithreading
- To learn socket programming concept

Course Outcomes:

- On completion of the course, student will be able to–
- To access open database through Java programs using Java Data Base Connectivity (JDBC) and develop the application.
- Understand and create dynamic web pages, using Servlets and JSP.
- Work with basics of framework to develop secure web applications

Course Contents:

Chapter-1	Chapter Name: Database Programming	Hours: 08

- The role of jdbc, The design of jdbc
- Types of drivers
- Steps of jdbc to access database
- Connectivity with database
- Create JDBC Statements – Statement, PreparedStatement, CallableStatement
- Scrollable and updatable result sets - TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE, TYPE_SCROLL_SENSITIVE - CONCUR_READ_ONLY, CONCUR_UPDATABLE

- 1.6 Metadata – DatabaseMetadata, ResultSetMetadata (Database : PostgreSQL)

Chapter-2	Chapter Name: Networking	Hours: 04

- The java.net package - InetAddress, URL, URLConnection class
- Connection oriented transmission – Stream Socket Class, SocketServer and Socket class
- Creating a Socket to a remote host on a port (creating TCP client and server)
- Simple Socket Program Example.
- Implementing Client-Server Applications

Chapter-3	Chapter Name: Multithreading	Hours: 06

- Introduction to Thread
- Life cycle of thread
- Thread Creation
  - By using Thread Class
  - By Using Runnable interface
- Priorities and Synchronization
- Running multiple thread
- Concurrency API: Executors, Locks, and Synchronizers
- Deadlocks and Thread Safe Design Patterns

Chapter-4	Chapter Name: Servlet	Hours: 06

- 4.1 Introduction to Servlet and Hierarchy of Servlet
- 4.2 Life cycle of servlet
- 4.3 Tomcat configuration (Note: Only for Lab Demonstration)
- 4.4 Handing get and post request (HTTP)
- 4.5 Handling a data from HTML to servlet
- 4.6 Retrieving a data from database to servlet
- 4.7 Session tracking – User Authorization, URL rewriting, Hidden form fields, Cookies and HTTP Session

Chapter-5	Chapter Name: JSP	Hours: 04

- Simple first JSP program
- Life cycle of JSP
- Implicit Objects
- Scripting elements – Declarations, Expressions, Scriplets, Comments
- JSP Directives – Page Directive, include directive
- Mixing Scriplets and HTML
- Example of forwarding contents from database to servlet, servlet to JSP and displaying it using JSP scriplet tag
- Introduction to JavaServer Faces (JSF)

Chapter-6	Chapter Name: Introduction to Frameworks	Hours: 02

- Spring
- Introduction of Spring framework, Bean
- Spring Applications
- Spring – MVC framework
- Introduction to Components of Hibernate

- Building a Simple CRUD Application with Spring and Hibernate

Reference Books:

- Core Java Volume I - Fundamentals By Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-0-13-516630-7
- The Complete Reference By Herbert Shildt, 11th Edition, McGraw Hill Education, ISBN 978-260-44023-2
- Java Beginners Guide By Herbert Shildt, 8 th Edition, McGraw-Hill Education ISBN 978-1- 260-44021-8
- Core Java Volume II – Fundamentals By Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-013-516631-4
- Java 2 Programming Black Book By Steven Holzner, DreamTech Press, ISBN 978-93-5119-953-4

E Books:

- The Complete Reference By Herbert Shildt https://gfgc.kar.nic.in/sirmv-science/GenericDocHandler/138-a2973dc6-c024-4d81-be6d-5c3344f232ce.pdf
- Java 2 Programming Black Book By Steven Holzner https://idoc.pub/documents/java-2-black-book-steven-holzner-vyly2rmq9v4m**,**

Evaluation Scheme

CIE : 15 Marks

SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-561-MJ-PR	Major Elective	Lab Based on CA-560-MJ-TH	Practical	02	04

Course Contents:

Chapter-1	**Chapter Name: Database Programming**

- Write a JDBC program to display all the details of the Person table in proper format on the screen. Create a Person table with fields as PID, name, gender, birth_year in PostgreSQL. Insert values in Person table.
- Write a program to display information about the ResultSet like number of columns available in the ResultSet and SQL type of the column. Use Person table. (Use ResultSetMetaData).
- Write a JDBC program to display all the countries located in West Region. Create a table Country in PostgreSQL with fields (Name, continent, Capital,Region). Insert values in the table.
- Write a JDBC program to insert the records into the table Employee(ID,name,salary) using PreparedStatement interface. Accept details of Employees from user.
- Write a JDBC program to perform search operation on Person table.
  1. Search all the person born in the year 1986.
  2. Search all the females born between 2000- 2005.
- Create an Employee Management System Using JDBC.
- Build a Library Management System with JDBC.

Chapter-2	**Chapter Name: Networking**

- Write a client-server program which displays the server machine's date and time on the client machine.
- Write a program to find primary IP address of the host name which you passed as a parameter

- Write a program which sends the name of a text file from the client to server and displays the contents of the file on the client machine. If the file is not found, display an error message.
- Write a program to accept a list of file names on the client machine and check how many exist on the server. Display appropriate messages on the client side.
- Write a server program which echoes messages sent by the client. The process continues till the client types "END".

Chapter-3	**Chapter Name: Multithreading**

- Write a multithreading program in java to display all the vowels from a given String. 2. Write a multithreading program using Runnable interface to blink Text on the frame.
- Write a program that create 2 threads – each displaying a message (Pass the message as a parameter to the constructor). The threads should display the messages continuously till the user presses ctrl-c. Also display the thread information as it is running.
- Write a java program to calculate the sum and average of an array of 1000 integers (generated randomly) using 10 threads. Each thread calculates the sum of 100 integers. Use these values to calculate average. [Use join method]
- Define a thread called "PrintText_Thread" for printing text on command prompt for n number of times. Create three threads and run them. Pass the text and n as parameters to the thread constructor. Example:
  i. First thread prints "I am in FY" 10 times
  ii. Second thread prints "I am in SY" 20 times
  iii. Third thread prints "I am in TY" 30 times
- Write a program to simulate traffic signal using threads
- Write a program to calculate prime numbers using concept of multithreading.

Chapter-4	**Chapter Name: Servlet**

- Write a servlet program to display current date and time of server.
- Design a servlet to display "Welcome IP address of client" to first time visitor. Display Welcome-back IP address of client" if the user is revisiting the page. (Use Cookies)(Hint: Use req.getRemoteAddr() to get IP address of client)
- Design the table User (username, password) using Postgre Database. Design HTML login screen. Accept the user name and password from the user. Write a servlet program to accept the login name and password and validates it from the database you have created. If it is correct then display Welcome.html otherwise display Error.html.
- Design a servlet that provides information about a HTTP request from a client, such as IP address and browser type. The servlet also provides information about the server on which the servlet is running, such as the operating system type, and the names of currently loaded servlets.
- Write a servlet which counts how many times a user has visited a web page. If the user is visiting the page for the first time, display a welcome message. If the user is re-visiting the page, display the number of times visited. (Use cookies).
- Create Dynamic User Registration and Authentication Servlet: A Comprehensive Web Application for Secure User Account Management with Input Validation, Session Handling, and Database Integration

Chapter-5	**Chapter Name: JSP**

- Write a Program to make use of following JSP implicit objects:
  i. out: To display current Date and Time.
  ii. request: To get header information.
  iii. response: To Add Cookie

iv. config: get the parameters value defined in
v. application: get the parameter value defined in
vi. session: Display Current Session ID
vii. pageContext: To set and get the attributes.
viii. page: get the name of Generated Servlet

- Create a JSP page which will accept the file extension and display all files in the current directory having that extension. Each filename should appear as a hyperlink on screen.
- Create a JSP page to accept a number from a user and display it in words: Example: 123 – One Two Three.
- Write a JSP program to perform Arithmetic operations such as Addition, Subtraction, Multiplication and Division. Design a HTML to accept two numbers in text box and radio buttons to display operations. On submit display result as per the selected operation on next page using JSP.
- Create a JSP page, which accepts user name in a text box and greets the user according to the time on server side. Example: If user name is Admin Output: If it is morning then display message in red color as, Good morning, Admin Today's date: dd/mm/yyyy format Current time: hh:mm:ss format If it is afternoon then display message in green color as, Good afternoon, Admin Today's date: dd/mm/yyyy format Current time: hh:mm:ss format If it is evening then display message in blue color as, Good evening, Admin Today's date: dd/mm/yyyy format Current time: hh:mm:ss format (Hint: To display date and time use GregorianCalendar and Calendar class)
- Write a JSP program to display number of times user has visited the page. (Use cookies)

Evaluation Scheme

CIE : 15 Marks

SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-562-MJ-TH	Major Elective	C# and .NET	Theory	02	02

Course Objectives:

- To understand development of windows application
- To learn data access mechanism.
- Create a web application
- Understand MVC Framework

Course Outcomes:

On completion of the course, student will be able to–

- Understand the VB.NET,C# and ASP
- Design and develop window based and web based .NET applications.
- Design and Implement database connectivity using ADO.NET

Course Contents:

Chapter-1	**Introduction to VB .NET**	**Hours: 08**

1.1 Basics of VB.Net
1.1.1 Operators
1.1.2 Data Types
1.1.3 Control Structures
1.2 Build Windows Applications
1.2.1 Controls: Form, TextBox, Button, Label, CheckBox, ListBox, ComboBox, RadioButton, ateTimePicker,
MonthCalender, Timer, Progressbar,Scrollbar, PictureBox, ImageBox, ImageList, TreeView, ListView,
Toolbar, StatusBar, Datagridview
1.2.2 Menus and PopUp Menu
1.2.3 Predefined Dialog controls: Color,Save,File,Open, Font

1.2.4 DialogBox - InputBox(), MessageBox, MsgBox()		
Chapter-2	**Introduction to C#**	**Hours: 07**
2.1. Language Fundamentals 2.1.1 Data type and Control Constructs 2.1.2 Value and Reference Types, Boxing 2.1.3 Arrays 2.1.4 String 2.1.5 Functions 2.2. Object Oriented Concepts 2.2.1 Defining classes and Objects 2.2.2 Access modifiers 2.2.3 Constructors 2.2.4 Inheritance 2.2.5 Interface 2.2.6 Abstract Class 2.2.7 Method Overloading and Overriding		
Chapter-3	**ASP .NET**	**Hours: 08**
3.1What isASP.NET? 3.2 Architecture ofASP.NET 3.3 Forms, WebPages, HTML forms 3.4 Request & Response in Non-ASP.NET pages 3.5 Using ASP.NET Server Controls 3.6 Overview of Control structures 3.7 Functions 3.8 Introduction to Web forms 3.8.1 Web Controls 3.8.2 Server Controls 3.8.3 Client Controls 3.8.4 Navigation Controls 3.8.5 Validations 3.8.6 Master Page		
Chapter-4	**ADO .NET and MVC**	**Hours: 07**
4.1Basics of Ado.net 4.1.1 Connection Object 4.1.2 Command Object 4.1.3 Dataset 4.1.4 Data Table 4.1.5 Data Reader Object 4.1.6 Data Adapter Object 4.2 Datagridview & Data Binding: Insert, Update, Delete records 4.3 Navigation Using Data Source 4.4 MVC Framework 4.4.1 Creating MVC Application 4.4.2 MVC File & Folder structure		
Evaluation Scheme **CIE : 15 Marks** **SEE : 35 Marks**		

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours per week
II	CA-563-MJ-PR	Major Elective	Lab Based on CA-562-MJ-TH	Practical	02	04

Course Contents:

Sample C#.NET Assignments:

1. Write a program to check whether the number is even or odd, print out an appropriate message to the user.
2. Write a program which will find all such numbers which are divisible by5.
3. Write a program which can compute the factorial of a given numbers.
4. Write a program that prints out all the elements of the list that are less than 10.
5. Write a program to determine whether the number is prime or not.
6. Write a program to check whether a number is palindrome or not. (using recursion and without recursion).
7. Write a C# program that reads a number from the user and calculates its square root. Handle the exception if the number is negative.
8. Write a C# program that prompts the user to input two numbers and divides them. Handle an exception when the user enters non-numeric values.
9. Write a C# Sharp program that takes three letters and displays them in reverse order.
10. Write a C# Sharp program that takes a character as input and checks if it is a vowel, a digit, or any other symbol.
11. Write a C# Sharp program to accept a person's height in centimeters and categorize them according to their height.
12. Write a C# Sharp program to read roll no, name and marks of three subjects and calculate the total, percentage and division.
13. Write a program in C# Sharp which is a menu-driven program to perform simple calculations.
14. Write a program in C# Sharp to create a function to input a string and count the number of spaces within the string.

15. Write a program in C# Sharp to calculate the sum of elements in an array.
16. Write a program in C# Sharp to create a recursive function to find the factorial of a given number.
17. How to interact with the user, with the Request.QueryString command.
18. Write a program to interact with the user, with the Request.Form command.
19. Write a program to interact with the user, through radio buttons, with the Request.Form command.
20. Write a program to create an open connection to a data source using the ADO Connection object. Through this connection, you can access and manipulate a database.

Evaluation Scheme

CIE : 15 Marks

SEE : 35 Marks

Restructured Syllabus (CBCS Pattern as per NEP 2020)

To be implemented from Academic Year: 2024-25

Faculty	**Science & Technology**
Program	**M.Sc. Computer Application**
Class	**F.Y.M.Sc.(Computer Application) Sem II**

Semester	Course Code	Type of Course	Course Title	Theory/ Practical	Credits	No. of clock hours
II	CA-581-OJT/FP	OJT/FP	Industry Internship / Field Project (FP)	Field Project	02	120

Course Objectives

- To provide students with an experience in working on projects or working within industry
- To inculcate Problem solving skills and work culture of the industry
- To foster team spirit
- To expose students with documentation used in industry

Course Outcomes

On Completion of this course, student will be able to –

- CO1: Make Use of tools used in industry
- CO2: Solve complex problems
- CO3: Effectively communicate and collaborate with team members and mentors.
- CO4: Demonstrate the ability to prepare documentation needed in the SDLC

Guidelines for Conduction of Industry Internship / Field Project

1. Faculty advisors / mentors shall decide whether a student shall work on industry internship or on a field project as per his/her plan/inclination at the beginning of the semester-II or earlier. The OJT may be carried out in physical or online form at the chosen industry.

2. Field Project should be strictly carried out under the guidance of the assigned faculty advisor / mentor. The assigned Faculty advisor / mentor shall monitor and track the OJT/FP

3. Internship / Field Project of 120 Hrs to be undertaken immediately after the end of SEM II examination and should be completed before the commencement of Semester III. However, Field Project may be undertaken during the semester II itself. 4. At the end of the industry internship / Field Project the student shall submit the report based on work undertaken during internship / Field Project as per prescribed format. 5. Student shall submit progress report on a periodic basis to Faculty advisor/ Mentor. Faculty advisor / mentor shall evaluate the work carried out by the student during internship / Field Project on a continuous basis for 30 marks. 6. The panel of examiners appointed shall evaluate the internship / Field Project based on submitted report and documentation for 70 marks.
Evaluation Scheme **CIE : 30 Marks** **SEE : 70 Marks**