Savitribai Phule Pune University

(Formerly University of Pune)

Two Year Masters Degree Program in Computer Science

(Faculty of Science and Technology)

Syllabi for M.Sc. (Computer Science) Part-I

(For Colleges Affiliated to Savitribai Phule Pune University)

Choice Based Credit System (CBCS) Syllabus
Under National Education Policy (NEP)

To be implemented from Academic Year 2023-2024

Title of the Course: M.Sc. (Computer Science)

Preamble

The Master of Science in Computer Science (M.Sc. CS) program is designed to provide advanced education and training in the field of Computer Science. This comprehensive program aims to equip students with a profound understanding of theoretical concepts, practical skills, and cutting-edge technologies relevant to the rapidly evolving world of computing.

With a strong emphasis on academic excellence and research-driven learning, the M.Sc. CS program seeks to nurture a community of skilled Computer Science professionals capable of addressing complex challenges across various industries. By fostering a stimulating and innovative learning environment, we strive to empower our students to become leaders, innovators, and agents of positive change in the field of Computer Science.

Eligibility

(a) Bachelor of Computer Science (B.C.S.) OR
(b) B.Sc.(Computer Science) OR
(c) B.C.A.(Science) OR
(d) B.Sc.(Information Technology) OR
(e) B.Sc.(Data Science) OR
(f) B.Sc.(Cyber and Digital Science) OR
(g) B.Sc. (Cyber Security) OR
(h) B.Sc. (Cloud Computing) OR
(i) Bachelor of Engineering(BE) in Computer Science/Information Technology/Electronics and Telecommunication/AI and Data Science/AI and Machine Learning/ equivalent OR
(j) B.Voc. in Software Development/ Information Technology
(k) B.Sc. with Computer Science as Principal Subject
(l) General B.Sc. with Computer Science as one of the subject at TYBSc level

Programme Outcomes:

PO 1: The Programme seeks to instill in students a deep and comprehensive knowledge of core computer science disciplines, advanced computer science concepts, theories, and principles, including algorithms, data structures, programming languages, artificial intelligence, machine learning, cloud computing, advanced databases, full stack development, software project management, and design patterns.

PO 2: Graduates should be equipped with the ability to analyze complex problems in computer science, design innovative solutions, and implement them effectively.

PO 3: The program aims to develop students' research skills, enabling them to evaluate existing research, contribute to knowledge in the field, and apply critical thinking to solve computational problems.

PO 4: The program aims to cultivate a passion for research, encouraging students to engage in original research projects that contribute to the advancement of computer science knowledge and address real-world problems.

PO 5: Students are expected to gain proficiency in multiple programming languages and develop the ability to write efficient, reliable, and maintainable code.

PO 6: Depending on the chosen track or concentration, students may develop expertise in areas.

PO 7: Through hands-on projects, practical assignments, and exposure to state-of-the-art tools and technologies, we aim to develop the technical proficiency and problem-solving skills necessary for success in the professional world.

PO 8: Graduates should be adept at presenting complex technical concepts clearly and effectively, both in written and oral forms, to various audiences.

PO 9: Computer science professionals often work in multidisciplinary teams. Students should learn to collaborate effectively with team members, understand different perspectives, and contribute productively to achieve common goals.

PO 10: The program places a strong emphasis on ethical considerations, responsible use of technology, and awareness of the societal impact of computing solutions. We aim to produce graduates who approach their work with integrity and a sense of social responsibility.

PO 11: Acknowledging the dynamic nature of computer science, we aim to instill in our students a desire for continuous learning and professional development, empowering them to adapt and thrive in the face of technological advancements; prepared them to adapt to new technologies and methodologies throughout their careers.

PO 12: Students will be encouraged to think creatively and innovatively, exploring new ideas and approaches to solve computational problems and advance the state of the art in the field.

PO 13: The program include On Job Training, internships, research work, research article and papers writing or a thesis that provides students with practical experience, applying their knowledge to real-world challenges.

The **Master of Science in Computer Science (M.Sc. CS)** program is committed to providing a rigorous and intellectually stimulating education that prepares graduates to excel in the ever-evolving field of computer science. The aim to nurture individuals who not only possess technical prowess but also demonstrate leadership, ethical conduct, and a dedication to making a positive impact on society through their knowledge and expertise.

Savitribai Phule Pune University

Syllabus Structure as per NEP Guidelines

M. Sc. (Computer Science) from 2023-24

SEMESTER I

Course Type	Course code	Course Name	Credits		Teaching Scheme Hrs/Week		Examination Scheme and Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core	CS-501-MJ	Advanced Operating System	4	-	4	--	30	70	100
	CS-502-MJ	Artificial Intelligence	4	-	4	--	30	70	100
	CS-503-MJ	Principles of Programming Languages	2	-	2	--	15	35	50
	CS-504-MJP	Lab course on CS-501-MJ	-	2	--	4	15	35	50
	CS-505-MJP	Lab course on CS-502-MJ	-	2	--	4	15	35	50
Major Elective	CS-510-MJ	Advance Databases and Web Technologies	2	-	2	--	15	35	50
	CS-511-MJP	Lab course on CS-510-MJ	-	2	--	4	15	35	50
	OR								
	CS-512-MJ	Cloud Computing	2	-	2	--	15	35	50
	CS-513-MJP	Lab course on CS-512-MJ	-	2	--	4	15	35	50
	OR								
	CS-514-MJ	C# .NET Programming	2	-	2	--	15	35	50
	CS-515-MJP	Lab Course on CS-514-MJ	-	2	--	4	15	35	50
RM	CS-531-RM	Research Methodology	4	-	4	--	30	70	100
		Total	**16**	**6**					

Savitribai Phule Pune University

Syllabus Structure as per NEP Guidelines

M. Sc. (Computer Science) from 2023-24

SEMESTER II

Course Type	Course code	Course Name	Credits		Teaching Scheme Hrs/Week		Examination Scheme and Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core	CS-551-MJ	Design and Analysis of Algorithms	4	-	4	--	30	70	100
	CS-552-MJ	Mobile App Development Technologies	4	-	4	--	30	70	100
	CS-553-MJ	Software Project Management	2	-	2	--	15	35	50
	CS-554-MJP	Lab course on CS-551-MJ	-	2	--	4	15	35	50
	CS-555-MJP	Lab course on CS-552-MJ	-	2	--	4	15	35	50
Major Elective	CS-560-MJ	Full Stack Development - I	2	-	2	--	15	35	50
	CS-561-MJP	Lab Course on CS-560-MJ	-	2	--	4	15	35	50
	OR								
	CS-562-MJ	Web Services	2	-	2	--	15	35	50
	CS-563-MJP	Lab Course on CS-562-MJ	-	2	--	4	15	35	50
	OR								
	CS-564-MJ	ASP.NET Programming	2	-	2	--	15	35	50
	CS-565-MJP	Lab course on CS-564-MJ	-	2	--	4	15	35	50
On Job Training	CS-581-OJT	On Job Training/Internship (120 Hours)	-	4	-	-	30	70	100
		Total	**12**	**10**					

Savitribai Phule Pune University

Syllabus Structure as per NEP Guidelines

M. Sc. (Computer Science) from 2023-24

SEMESTER III

Course Type	Course code	Course Name	Credits		Teaching Scheme Hrs/Week		Examination Scheme and Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core	CS-601-MJ	Software Architecture and Design Pattern	4	-	4	--	30	70	100
	CS-602-MJ	Machine Learning	4	-	4	--	30	70	100
	CS-603-MJ	Internet of Things	2	-	2	--	15	35	50
	CS-604-MJP	Lab course on CS-601-MJ and 603	-	2	--	4	15	35	50
	CS-605-MJP	Lab course CS-602-MJ	-	2	--	4	15	35	50
Major Elective	CS-610-MJ	Full Stack Development- II	2	-	2	--	15	35	50
	CS-611-MJP	Lab Course on CS-610-MJ	-	2	--	4	15	35	50
	OR								
	CS-612-MJ	DevOps Fundamentals	2	-	2	--	15	35	50
	CS-613-MJP	Lab Course on CS-612-MJ	-	2	--	4	15	35	50
	OR								
	CS-614-MJ	Soft Computing	2	-	2	--	15	35	50
	CS-615-MJP	Practical on CS-614-MJ	-	2	--	4	15	35	50
Research Project	CS-631-RP	Research Project Work (120 Hrs)	-	4	-	-	30	70	100
		Total	**12**	**10**					

Savitribai Phule Pune University

Syllabus Structure as per NEP Guidelines

M. Sc. (Computer Science) from 2023-24

SEMESTER IV

Course Type	Course code	Course Name	Credits		Teaching Scheme Hrs/Week		Examination Scheme and Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Core	CS-651-MJP	Full Time Industrial Training (IT)	-	12	-	-	90	210	300
Major Elective	CS-652-MJ	Online/MOOC (Elective Courses List)	4	-	-	-	30	70	100
Research Project	CS-681-RP	Research Project Work (180 hrs)	-	6	-	-	45	105	150
		Total	**4**	**18**					

Abbreviations

CS	Computer Science		**MJ**	Major Theory
RM	Research Methodology		**MJP**	Major Practical
OJT	On Job Training		**RP**	Research Project
TH	Theory		**PR**	Practical
CE	Continuous Evaluation		**EE**	End semester Evaluation
MOOC	Massive Open Online Course			

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-501-MJ : Advanced Operating System

No. of Credits: 4	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 30 Marks End Semester : 70 Marks

Prerequisite

- Working knowledge of C programming.
- Basic Computer Architecture concepts.
- Basic algorithms and data structure concepts

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 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

CO7: Understand the memory management policies of UNIX/LINUX OS

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to UNIX/Linux Kernel**	**5**	**CO1**

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

[Book-1]

2	**Unix/Linux File Subsystem**	**8**	**CO2**

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

[Book-1]

3	**System Calls for File Subsystem**	**12**	**CO3**

3.1 File I/O System calls

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

[Book-1]

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

[Book-2]

4	**Unix/Linux Process Control Subsystem**	**12**	**CO4**

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

[Book-1]

5	**System Calls Process Control Subsystem**	**8**	**CO5**

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

[Book-2]

6	**Signal Handling**	**7**	**CO6**

7.1 Introduction

7.2 Signal Concepts

7.3 Signal function

7.4 kill and raise functions

7.5 alarm and pause functions

7.6 abort function

7.7 sleep function

			[Book-2]
7	**Memory Management**	**8**	**CO7**
6.1 Swapping - Allocation of swap space, Swapping process out, Swapping process in 6.2 Demand Paging - Data structures for demand paging, Page stealer process, Page faults			[Book-1]
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			

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-502-MJ : Artificial Intelligence

No. of Credits: 4	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 30 Marks End Semester : 70 Marks

Prerequisite

- Data Structure and Algorithm.
- Discrete mathematics.
- Knowledge of Programming Language
- Data Analytics Skill

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.

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to Artificial Intelligence**	**6**	**CO1, CO2**
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.			
2	**Problem Solving**	**10**	**CO2, CO3**
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*.			
3	**Game Theory**	**10**	**CO3, CO4**
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).			
4	**Knowledge Representation**	**10**	**CO2, CO4**
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.			
5	**Reasoning**	**10**	**CO4, CO5**
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			
6	**Planning**	**8**	**CO5, CO6**
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			
7	**Recent trends in AI**	**6**	**CO7**
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 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

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-503-MJ : Principles of Programming Language

No. of Credits: 2	Teaching Scheme Theory: 2 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Procedural Language like C
- Object-Oriented Languages
- Basic Data Structures and Algorithms

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

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction**	**2**	**CO1**

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

2	**Names, Scopes, Bindings, Object Orientation Concepts**	**6**	**CO1, CO2**

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

3	**Data Types**	8	**CO2, CO3**

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

4	**Control Flow**	6	**CO2,CO3**

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

5	**Subprograms and Implementing Subprograms**	8	**CO3,CO4**

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

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-504-MJP : Lab Course on CS-501-MJ (Advanced Operating System)

No. of Credits: 2	**Teaching Scheme** Theory: 4 Hrs/Week	**Examination Scheme** Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Working knowledge of C programming.
- Basic Computer Architecture concepts.
- Basic algorithms and data structure concepts

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

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.

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-505-MJP : Lab Course on CS-502-MJ (Artificial Intelligence)

No. of Credits: 2	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Python Programming Language
- Data Structure and Algorithm.
- Discrete mathematics.
- Knowledge of Programming Language
- Data Analytics Skill

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.

Assign No.	Practical Assignment
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 (a) The lift would halt at a particular floor based on the identity of the individual. (b) 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)

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-510-MJ : Advance Databases and Web Technologies

No. of Credits: 2	Teaching Scheme Theory: 2 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Knowledge of file system concepts
- A firm foundation of any RDBMS package
- Knowledge of Database Concepts
- Basic knowledge of HTML and CSS
- Basic knowledge of JavaScript.
- Basics of web application development.
- Knowledge of what is Clint and Server-side programming.

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

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to NOSQL**	**5**	**CO1**
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			
2	**NOSQL Databases**	**9**	**CO1, CO2**
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

3	**Basics of HTML5**	4	**CO2, CO3**

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, height and width, list, min and max, multiple, pattern (regexp)

4	**CSS3 Introduction**	4	**CO2, CO3**

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

5	**Introduction to BootStrap**	8	**CO3, CO4**

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
6. Responsive Web Design with HTML5 and CSS: Develop Future-proof Responsive Websites Using the Latest HTML5 and CSS Techniques Book by Ben Frain
7. Bootstrap 4 Quick Start: A Beginner's Guide to Building Responsive Layouts with Bootstrap 4 Book by Jacob Lett
8. Bootstrap: Responsive Web Development Book by Jake Spurlock

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-511-MJP : Lab Course on CS-510-MJ (Advance Databases and Web Technologies)

No. of Credits: 2	**Teaching Scheme** Theory: 4 Hrs/Week	**Examination Scheme** Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Knowledge of file system concepts
- A firm foundation of any RDBMS package
- Knowledge of Database Concepts
- Basic knowledge of HTML and CSS
- Basic knowledge of JavaScript.
- Basics of web application development.
- Knowledge of what is Clint and Server-side programming

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

Assign No.	**Name of 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 thecount 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 a. Date time b. email input type c. search input type 15. Write an HTML 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

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-512-MJ : Cloud Computing

No. of Credits: 2	Teaching Scheme Theory: 2 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Operating System
- Fundamentals of Computer Networks
- Good Understanding of Object Oriented Programming Concepts

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

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to Cloud Computing**	**8**	**CO1**

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		
2	**Abstraction and Virtualization**	**5**	**CO2**
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			
3	**Overview of Cloud Security**	**8**	**CO3**
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.			
4	**Cloud Technologies and Advancements**	**9**	**CO4**

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

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-513-MJP: Lab Course on CS-512-MJ (Cloud Computing)

No. of Credits: 2	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Operating System
- Fundamentals of Computer Networks
- Good Understanding of Object Oriented Programming Concepts

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.

Assign No.	Name of 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.

Reference Books
4. Brian J.S. Chee and Curtis Franklin : Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center 5. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi : Mastering Cloud Computing: Foundations and Applications Programming 6. Kai Hwang, Geoffrey C Fox, Jack G Dongarra : Distributed and Cloud Computing, From Parallel Processing to the Internet of Things

SavitribaiPhulePuneUniversity
M.Sc. Computer Science (2023)

CS-514-MJ : C# .NET Programming

No. of Credits: 2	Teaching Scheme Theory:2 Hrs/Week	Examination Scheme ContinuousEvaluation:15Marks End Semester:35Marks

Prerequisite

- Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
- Familiarity with programming language such as C++ and/or Java.

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#

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to .Net Framework**	**2**	**CO1**
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),			
2	**Introduction to C#.Net**	**4**	**CO1**
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			
3	**OOPS with C#**	**5**	**CO3**
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			
4	**Data Structure**	**2**	**CO3**
4.1 ArrayList 4.2 Collection 4.3 Dictionary 4.4 Hash Table			
5	**Multithreading I/O Stream**	**3**	**CO4**
5.1 Stream Reader, Stream Writer 5.2 File Mode 5.3 Opening & Closing File 5.4 Random Access File			
6	**Assembly Components**	**2**	**CO1**
6.1 .NET Assembly features 6.2 Structure of Assemblies 6.3 Calling assemblies, private and shared assemblies			
7	**Windows Programming**	**6**	**CO5**
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			
8	**Database Connectivity using ADO.NET**	**6**	**CO6**
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 References
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**

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-515-MJP : Lab Course on CS-514-MJ (C# .NET Programming)

No. of Credits: 2	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
- Familiarity with programming language such as C++ and/or Java.

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.

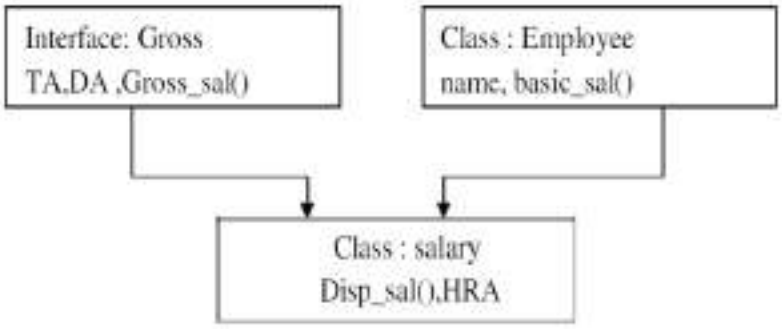
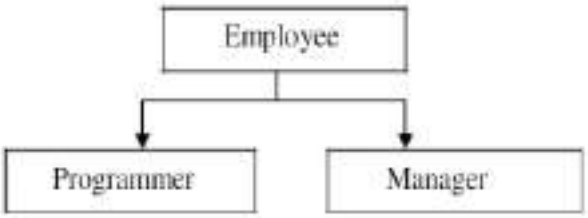
CO3: Design & implement Object Oriented Programming concepts like Inheritance and Polymorphism in C# programming language.

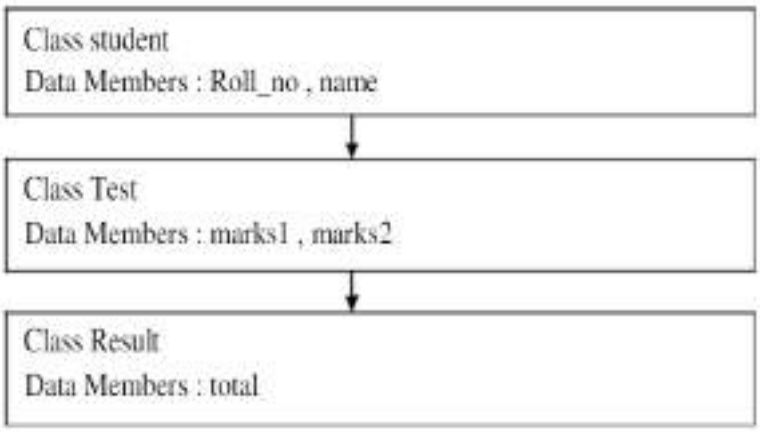
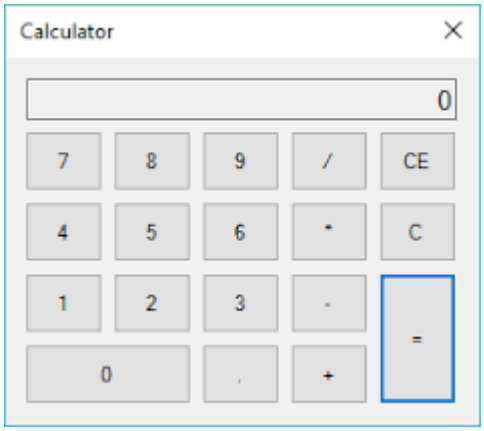
CO4: Design & Implement the application using multithreading & File handling

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#

Assign No.	Name of 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 Write a C# program to accept one string &character , find the occurrence of char from string using function
11-19	**OOPs Concepts:** 10. 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. 11. Write a program to swap three integer and three float numbers using the concept of Function overloading. 12. 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. 13. C# program to demonstrate the example of multilevel inheritance. 14. 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. 15. 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. 16. Program to implement the following multiple inheritance using interface.  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.

<table>
<tr><td></td><td> 
</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>
<tr><td>31-32</td><td>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
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.</td></tr>
<tr><td>33-37</td><td>Windows Programming
32. Create a windows application to perform following basic arithmetic operations
 

33. 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.</td></tr>
</table>

	34. 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. 35. 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. 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:** 36. Write a C# application using ADO.NET to verify if the connection is established with the database or not. Display appropriate messages 37. Write a C# application using ADO.NET to perform insert, delete, update and select operation. 38. 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. 39. 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. 40. 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 ___.

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-531-RM : Research Methodology

No. of Credits:	Teaching Scheme	Examination Scheme
4	Theory: 4 Hrs/Week	Continuous Evaluation: 30 Marks End Semester : 70 Marks

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.

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to Research Methodology**	**10**	**CO 1,9**
	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		
2	**Literature Review and Formulation of Research Problems**	**6**	**CO 1,2,3**
	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		
3	**Research Design**	**8**	**CO 3,4**
	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		
4	**Hypothesis and Sampling**	**10**	**CO 5,6**
	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		
5	**Data Collection, Processing and Analysis of Data**	**10**	**CO 5**
	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		
6	**Interpretation and Report Writing**	**8**	**CO 6,7,8**
	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		
7	**Publication Ethics and Open Access Publishing**	**8**	**CO 7,9**
	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
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.

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-551-MJ : Design and Analysis of Algorithms

No. of Credits: 4	Teaching Scheme Theory:4 Hrs/Week	Examination Scheme ContinuousEvaluation:30 Marks End Semester:70 Marks

Prerequisite

- Basic knowledge of algorithms and programming concepts
- Data Structures and Advanced Data Structures
- Basic Knowledge of Graphs and Algorithms

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.

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Basics of Algorithms**	**6**	**CO1, CO2, CO5**
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			
2	**Divide and Conquer strategy**	**10**	**CO2, CO3, CO5**
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, mini-max algorithm, largest number multiplication, simple convex algorithm			
3	**Greedy Method**	**10**	**CO2, CO3,CO5**
3.1 Knapsack problem 3.2 Job sequencing with deadlines 3.3 Minimum-cost spanning trees: Kruskal and Prim's algorithm 3.4 Optimal storage on tapes 3.5 Optimal merge patterns 3.6 Huffman coding 3.7 Shortest Path :Dijkstra's Algorithm			
4	**Dynamic Programming**	**12**	**CO2, CO3,CO5**
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 All pairs Shortest Path Floyd-Warshall Algorithm 4.6 Travelling Salesperson problem 4.7 Longest common subsequence 4.8 String editing			
5	**Decrease and Conquer**	**6**	**CO2, CO4,CO5**
5.1 Definition of Graph Representation of Graph 5.2 By Constant - DFS and BFS 5.3 Topological sorting 5.4Strongly Connected components and spanning trees 5.5 Articulation Point and Bridge edge			
6	**Backtracking**	**7**	**CO2, CO3,CO5**
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			
7	**Branch and Bound Technique**	**6**	**CO3, CO4,CO5**
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)			
8	**Problem Classification**	**3**	**CO1,CO2, CO5**
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 8.3 Cook's theorem			
Reference Books			
4. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI Learning Private Limited, 2012.			

5. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Data Structures and Algorithms, Pearson Education, Reprint 2006.
6. Harsh Bhasin, Algorithms Design and Analysis, Oxford university press, 2016.
7. 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

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-552-MJ : Mobile App Development Technologies

No. of Credits:	Teaching Scheme	Examination Scheme
4	Theory: 4 Hrs/Week	Continuous Evaluation: 30 Marks End Semester : 70 Marks

Prerequisite

- Concepts of Networking and Conversant with OS internals
- Familiar with the java programming will be an added advantage
- Knowledge about different mobile platform

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:

On Completion of this course, student will be able -

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 GPS, accelerometer, Bluetooth, WiFi, Media Player and Camera, in their applications.
CO 7. To Get knowledge of Phone Gap Programming

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction Mobile Technologies**	**3**	**CO1**
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.		
2	**Fundamentals of Android Programming**	**6**	**CO1 CO2**
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		
3	**Android Activity, Intents, and Services**	**6**	**CO2, CO3**
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		
4	**Android UI Layouts and Controls for GUI Design**	**12**	**CO2, CO3**
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		
5	**Android Menus, Threads, Notification and Alarms**	**8**	**CO3, CO4**
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		
6	**Android ContentProviders, Broadcast Receivers and Parsing**	**8**	**CO4, CO5**
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		
7	**Advanced Android Programming**	**9**	**CO5, CO6**
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, Android Sensors		
7.6.	Android - Facebook Integration, Android - Gestures		

7.7. Publishing Android Application			
8	**Phone Gap Programming**	**8**	**CO7**
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 8.6. Phone Gap APIs –Accelerometer 8.7. Querying Device Orientation, watching a Device's Orientation 8.8. Creating Contacts, Searching for Contacts, Cloning Contacts, and Removing Contacts.			

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

Web References:

1. https://developer.android.com/guide
2. https://www.openxcell.com/mobile-app-development/
3. https://magora-systems.com/mobile-software-development-for-newbies/
4. https://www.apogaeis.com/blog/mobile-application-development-top-10-factors-to-consider/
5. https://www.ibm.com/topics/mobile-application-development
8. https://www.tutorialspoint.com/phonegap/index.htm

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-553-MJ: Software Project Management

No. of Credits: 2	Teaching Scheme Theory: 2 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Knowledge of Software Engineering
- Basic of software testing concepts

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

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

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.

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to Project Management**	**4**	**CO1**
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			
2	**Project Management Components**	**4**	**CO2**
2.1 Project Integration Management-Project plan 2.2 development and execution 2.3 Change controls and CCB 2.4 Configuration management			
3	**Scope, Time and Cost Management**	**6**	**CO2, CO3**
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 Cost estimation and Control 3.8 COCOMO model 3.9 BASIC COCOMO NUMERICALS			
4	**Quality Management and Quality Standards**	**4**	**CO3, CO4**
4.1 Quality planning and assurance 4.2 CMM levels 4.3 KPA's 4.4 PSP/TSP			
5	**Human Resource Management and Communication Management**	**4**	**CO2, CO4**
5.1 Organizational planning 5.2 Staff acquisition 5.3 Information distribution 5.4 Reporting			
6	**Risk and Procurement Management**	**4**	**CO2, CO3**
6.1 Risk identification 6.2 Quantification and control 6.3 Solicitation management and control 6.4 Contract administration.			
7	**Stakeholder Management and Software Metrics**	**4**	**CO1, CO3, CO4**
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 7.7 Analyzing software data			
Reference Books			
4. The Software Development Project: Planning and Management by Phillip Bruce and Sam M Pederson 5. Software Project Management : A Process-Driven Approach by Ashfaque Ahmed 6. Software Engineering Project Management by Richard Thayer, Edward Yourdon WILEY. 7. Introduction to Software Project Management by Adolfo Villafiorita CRC Press 8. Software Engineering by Roger Pressman McGraw-Hill 9. Software Metrics for Project Management and process improvement by Robert B. Grady Prentice hill			

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-554-MJP : Lab Course on CS-551-MJ (Design and Analysis of Algorithms)

No. of Credits: 2	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Basic knowledge of algorithms and programming concepts
- Data Structures and Advanced Data Structures
- Basic Knowledge of Graphs and Algorithms
- Basic knowledge of C/C++/ Java

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.

Assign No.	Name of Practical Assignment
1	Write programs in C/C++/ Java to sort a list of n numbers in ascending order using selection sort, insertion sort, heap sort, radix sort. Determine the time required to sort and compare on basis of time complexity for different values of n.
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 show board configuration of 4 queens problem
22	Write a program in C/C++/ Java to find out longest common subsequence from the given strings
23	Write a program in C/C++/ Java to find out live node, E node and dead node from a given graph
24	Write a program in C/C++/ Java to find out solution for travelling salesman problem using LCBB from a given matrix.
25	Write a program in C/C++/ Java to find out solution for 0/1 knapsack problem

Savitribai Phule Pune University

M.Sc. Computer Science (2023)

CS-555-MJP : Lab Course on CS-552-MJ (Mobile App Development Technologies)

No. of Credits: 2	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Concepts of Networking
- Conversant with OS internals
- Familiar with the network Protocol stack
- Knowledge about different mobile platform and application development
- Concept of wireless communication

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.

Practical Assignment Based on

- Simple Android Applications
- Android Activity, Intents, and Services
- Android GUI Design Applications
- Android Menus, Threads, Notification and Alarms
- Android Content Providers, Broadcast Receivers and Parsing
- Advanced Android Programming –SMS, MMS, Phone Call, Email, Bluetooth, WiFi, Camera, Media Player, Facebook Integration, GMap, Location base Service, etc.
- Phone Gap Programming

Assign No.	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	Digital Bio Data PhoneGap Application using HTML5.
27	Write a PhoneGap application to display push notification.
28	Write a PhoneGap application to create a contact, Searching for Contacts, Cloning Contacts, Removing Contacts.

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-560-MJ : Full Stack Development-I

No. of Credits: 2	Teaching Scheme Theory: 2 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Knowledge of HTML, CSS, JavaScript basics and MongoDB

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.

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to MEAN Stack**	**2**	**CO1**
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			
2	**Advanced ES6 features in JavaScript and Typescript**	**8**	**CO2**
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 Object-oriented programming 2.4.1 Classes 2.4.2 Inheritance 2.4.3 Encapsulation 2.4.4 Polymorphism 2.5 TypeScript 2.5.1 What is TypeScript? 2.5.2 Benefits of using TypeScript 2.5.3 Installing TypeScript 2.5.4 Writing TypeScript code 2.5.5 Types in TypeScript Basic types, Enums, Interfaces, Classes, Generics 2.6 Advanced TypeScript 2.6.1 Modules 2.6.2 Decorators 2.6.3 Type narrowing 2.6.4 Type guards			
3	**AngularJS**	**5**	**CO3**
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			
4	**Node.js**	**5**	**CO4**
4.1 Introduction to Node.js 4.2 Event loop 4.3 Asynchronous programming 4.4 Modules 4.5 Packages 4.6 Streams			
5	**ExpressJS**	**5**	**CO5**
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			
6	**Building a MEAN Stack Application**	**5**	**CO6**
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

Web Links

1. http://es6-features.org/
2. https://www.typescriptlang.org/
3. https://angular.io/
4. https://expressjs.com/
5. https://nodejs.org
6. https://www.w3schools.com/
7. https://www.tutorialspoint.com
8. https://www.tutorialsteacher.com/
9. https://www.geeksforgeeks.org/
10. https://www.javatpoint.com/
11. https://www.codeproject.com/

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-561-MJP : Lab Course on CS-560-MJ (Full Stack Development-I)

No. of Credits: 2	**Teaching Scheme** Theory: 4 Hrs/Week	**Examination Scheme** Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Knowledge of HTML and CSS basics

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

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

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-562-MJ : Web Services

No. of Credits: 2	Teaching Scheme Theory: 2 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Strong knowledge about Java programming.
- Good Understanding of Object Oriented Programming concepts.
- Must be familiar with XML

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.

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to Web Services**	**5**	**CO1**
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.			
2	**Web Services Architecture**	**4**	**CO2**
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.			
3	**SOAP: Simple Object Access Protocol**	**5**	**CO3**
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.

4	**Describing, Registering and Discovering Web Services**	**11**	**CO4**

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

5	**The REST Architectural Style**	**5**	**CO5**

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

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-563-MJP : Lab Course on CS-562-MJ (Web Services)

No. of Credits: 2	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester : 35 Marks

Prerequisite

- Strong knowledge about Java programming.
- Good Understanding of Object Oriented Programming concepts.
- Must be familiar with XML

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.

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.

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-564-MJ : ASP .NET Programming

No. of Credits: 2	Teaching Scheme Theory:2 Hrs/Week	Examination Scheme Continuous Evaluation:15Marks End Semester Exam:35Marks

Prerequisite

- Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
- Familiarity with programming language such as C++ and/or Java.

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

Unit No.	Name of Unit	Teaching Hours	CO Targeted
1	**Introduction to ASP.NET**	**2**	**CO1**
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			
2	**ASP.NET Sever controls**	**7**	**CO1**
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,			
3	**Manage state in ASP.NET**	**3**	**CO3**
3.1 View state, 3.2 Session state, 3.3 Application state, 3.4 Use of cookies and URL encoding			
4	**Web forms in ASP.NET**	**3**	**CO3, CO5**
4.1 Creating a web page, 4.2 create and develop content page, 4.3 Access web page controls from content page			
5	**Database connection programming in ASP.NET**	**7**	**CO5**
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			
6	**Debugging and Error handling in ASP.NET page level**	**3**	**CO2**
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			
7	**Setup and deploy web applications of ASP.NET**	**3**	**CO5**
7.1 Download and install IIS, 7.2 Deploy website in IIS, 7.3 Publishing ASP.NET website, 7.4 Unit testing			
8	**ASP.NET MVC**	**2**	**CO6**
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			

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-565-MJP : Lab Course on CS-564-MJ (ASP .NET Programming)

No. of Credits: 2	Teaching Scheme Theory: 4 Hrs/Week	Examination Scheme Continuous Evaluation: 15 Marks End Semester Exam : 35 Marks

Prerequisite

- Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
- Familiarity with programming language such as C++ and/or Java.

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

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.

Savitribai Phule Pune University
M.Sc. Computer Science (2023)

CS-581-OJT : On Job Training (Internship)

No. of Credits: 4	Total Duration 120 Hours	Examination Scheme Continuous Evaluation: 30 Marks End Semester : 70 Marks

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

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