DPU

Dr. D. Y. PATIL VIDYAPEETH, PUNE

(Deemed to be University)

Syllabus for Master of Computer Applications

(First Year 2023-24)

INDEX

REGULATION FOR THE MASTER OF COMPUTER APPLICATION MCA (2023-24)

1. **Eligibility**

   Bachelor's degree from any recognized university with an aggregate of 45% marks in aggregate (candidate belonging to open category and 40 % candidate belonging to any reserved category). Maths must be compulsory subject for the candidate seeking admission in MCA either in 12th or bachelor degree.

2. **Duration of the course**

   The MCA program is of TWO years (Total FOUR semesters) degree program.
   Duration of the course: 2 years i.e. 4 semesters. Semesters - An academic year consists of two semestersOdd Semester: June/July to November/December
   Even Semester: November/December to April/May

3. **Medium of instruction:**

   English shall be the medium of instruction for all the subjects of study andfor examination of the course.

4. **Attendance:**

   A candidate has to secure minimum-
   75% attendance in theory
   80% in practical for qualifying to appear for the final examination.

5. **Scheme of Examination**

Internal Examinations (Theory + Practical + Project)

1. There shall be two internal examinations (also called internal assessment tests I and II) of one hour duration for each course to be held as per the schedule fixed in the Academic Calendar.
2. A student can take for supplementary re-internal exam of a specific subjector all the subjects for the betterment of performance in case of scoring of less mark in previous internal assessment exams only after successful submission of an application to the class teacher which will be approved by Director/Principal of the institute.
3. Project and Seminar will be evaluated on the basis of 50% internal assessment and 50% end semester assessment in the form of project demonstration and PPT.
4. Value added courses (VA) and ability/skill enhancement courses(AEC) will be evaluated through the continuous internal assessment(CIA) will be graded.

b). University Examination (Theory)

University Theory Examination Pattern		
Section A		
MCQs	15 x 1 Mark each	15 Marks
Section B		
1)Very short and short Qs (Any 5out of 7) 2)short question (Any 2 out 0f 3)	03 x 05 Marks each 02 x 05 Marks each	15 Marks 10 Marks
Section C		
Long Questions (Any 2 out of 3)	02 x 10 Marks each	20 Marks
	Total	**60 Marks**

Total 100 Marks Combined Head of Passing

1. External Theory will carry 60 marks
2. Internal Assessment (Theory) will carry total of 40 marks

Practical Examination scheme

1. External Practical will carry 60% marks
2. Internal Assessment (Practical) will carry total of 40 % marks

Break –Up

1. Final Theory University Exams 60 Marks
2. Internal Assessment Exams 40 Marks
3. Grand Total = 100 Marks (Each Subject)

Note: for any subject examination scheme will be

Internal exam/evaluation for Theory and lab : 40 %
(Unit Test 1, Unit Test 2 and continuous assessment over the semester)
External exam/evaluation for Theory and lab : 60%

(c) Standard of Passing:

1. The standard of passing shall be minimum 50% in each subject.
2. The marks of all heads combined (University Theory Exam + Internal Assessment Theory + Practical / Viva) shall be considered together for Passing of the candidate.

(d) Grace Marks

The grace marks up to a maximum of 1 percentage of total marks may be awarded to a student who has failed in not more than two subjects in the respective semester. Provided that these grace marks shall be awarded onlyif the student passes after awarding these marks.

(e) Grading System

UGC 10-point Grading Scale

Marks	Letter Grade	Grade Point
90 To 100	**O** : Outstanding	10
80 To 89	**A+** : Excellent	9
70 To 79	**A** : Very Good	8
60 To 69	**B+** : Good	7
55 To 59	**B** : Average	6
50 To 54	**P** : Pass	5
00 To 49	**F** : Fail	0
-	**AB** : Absent	0

Computation of SGPA and CGPA

The UGC recommends the following procedure to compute the Semester Grade Point Average (SGPA) and Cumulative Grade Point Average (CGPA):

i. The SGPA is the ratio of sum of the product of the number of credits withthe grade points scored by a student in all the courses taken by a student and the sum of the number of credits of all the courses undergone by a student, i.e.
SGPA (Si) = Σ(Ci x Gi) / ΣCi
where Ci is the number of credits of the course and Gi is the grade point scored by the student in the course.

ii. The CGPA is also calculated in the same manner taking into account all the courses undergone by a student over all the semesters of a programme,i.e.
CGPA = Σ(Ci x Si) / Σ Ci where Si is the SGPA of the semester and Ci isthe total number of credits in that semester.

iii. The SGPA and CGPA shall be rounded off to 2 decimal points and reported in the transcripts.
Illustration of Computation of SGPA and CGPA and Format for Transcripts

(i) Criteria for appointment of Examiner (Internal & External) andterms of their appointment.

1. Adhoc Board of Studies of Computer Science and Engineering shallsubmit, to the Committee constituted by Board of Examinations, a panel of examiner names, along with their addresses, suitable for appointment asInternal and External Examiners.

2. Examiners shall be appointed by the Academic Council as per section 8(b)
(viii) of the Rules of Dr. D. Y. Patil University on the recommendations ofthe Board of Examinations.

3. In case of refusal from the person so appointed, the Controller of Examinations shall appoint substitute examiners from the panel approved.

4. Internal and External Examiners: An "Internal Examiner" means a personwho is a teacher in the constituent college(s) / institute(s) of the University.The teachers in other universities or recognized teacher of other Universityin the state or outside the state shall be referred to as the “External Examiner".

5. Intimation of appointment as the examiner shall be accompanied by a copy of the instructions/guidelines relating to the examination for he/she is appointed, as also the information regarding the remuneration he/she shallbe entitled to draw, if he/she acts as examiner. He/ She is expected to attendto and shall be required to send to the Controller of Examinations.

6. Examiners shall be appointed for examinations to be held in that academicyear; however they shall be eligible for reappointment.

7. Relatives, Close Friends or next to the kin which are directly or indirectlyrelated to the candidates shall not to be included.

Dr D. Y. Patil School of Science &Technology,
Dr D. Y. Patil Vidyapeeth, Pimpri, Pune
MCA

Program Outcomes (POs)

Learners are expected to know and be able to–

PO1	**Engineering knowledge**	Apply the knowledge of mathematics, science, Engineering fundamentals, and an Engineering specialization to the solution of complex Engineering problems.
PO2	**Problem analysis**	Identify, formulate, review research literature and analyze complex Engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and Engineering sciences.
PO3	**Design / Development of Solutions**	Design solutions for complex Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and Environmental considerations.
PO4	**Conduct Investigations of Complex Problems**	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern Tool Usage**	Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations.
PO6	**The Engineer and Society**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practices.
PO7	**Environment and Sustainability**	Understand the impact of the professional Engineering solutions in societal and Environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**	Apply ethical principles and commit to professional ethics and responsibilities and norms of Engineering practice.
PO9	**Individual and Team Work**	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication Skills**	Communicate effectively on complex Engineering activities with the Engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project Management and Finance**	Demonstrate knowledge and understanding of Engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary Environments.

PO12	**Life-long Learning**	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

(PSO)

A graduate of the Computer Engineering Program will demonstrate-

PSO 1	**Professional Skills-**The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexities.
PSO 2	**Problem-Solving Skills-** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO 3	**Successful Career and Entrepreneurship-** The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur and to have a zest for higher studies.

COURSE STRUCTURE FOR

MASTER OF COMPUTER APPLICATION (MCA)

SEMESTER I

Course Code	Course Type as per NEP	Course Name	L	T	P	Hr	Cr
MCA-CA-101	MAJOR	Mathematical Foundations of Computer Science	3	1	0	4	4
MCA-CA-102	MAJOR	OOP using Java	3	0	4	7	5
MCA-CA-103	MAJOR	Data Structures & Algorithms	3	0	4	7	5
PEC-CA-101	DSE	Discipline Specific Elective	3	0	2	5	4
PCC-CA-101	VA	Value Added	1	0	2	3	2
HSMC-CA-101	AEC	Ability Enhancement Course	2	0	0	2	2
		Total	15	1	12	27	22

VA: Technical English
DSE: Design Patterns / Software Architecture/Agile Methodology
AEC: Yoga/ Human Values & Ethics- 1/ Foreign Language -I(French/German/Japanese)

SEMESTER II

Course Code	Course Type as per NEP	Course Name	L	T	P	Hr	Cr
MCA-CA-201	MAJOR	Advanced JAVA	3	0	4	7	5
MCA-CA-202	MAJOR	Artificial Intelligence	3	0	2	5	4
MCA-CA-203	MAJOR	Database Management System	3	0	4	7	5
PEC-CA-201	DSE	Discipline Specific Elective	3	0	2	5	4
PCC-CA-201	VA	Value Added	1	0	2	3	2
HSMC-CA-201	AEC	Ability Enhancement Course	2	0	0	2	2
		Total	15	0	14	28	22

VA: Research Methodology & IPR
DSE: Front end development with HTML5, CSS3/Javascript / ReactJS/Angular
AEC:Sports/ Human Values & Ethics-II/ Foreign Language-II (French/German/Japanese)

SEMESTER III

Course Code	Course Type as per NEP	Course Name	L	T	P	Hr	Cr
MCA-CA-301	MAJOR	Introduction to Data Science	4	0	0	4	4
MCA-CA-302	MAJOR	Advanced Data Structure	3	0	4	7	5
MCA-CA-303	MAJOR	Data Communication And Networks	3	0	2	5	4
PEC-CA-301	DSE	Discipline Specific Elective	3	0	4	7	5
PCC-CA-301	VA	Value Added	1	0	2	3	2
HSMC-CA-301	AEC	Ability Enhancement Course	2	0	0	2	2
		Total	16	0	12	28	**22**

VA: Computer Assembly & Repair
DSE: DevOps/Cloud (AWS/AZURE)/Salesforce
AEC:Physical Education/ NSS/NCC/Internship/Apprenticeship

SEMESTER IV

	Course Type as per NEP	Course Name	L	T	P	Hr	Cr
PCC-CA-401	MAJOR	Research Project / CAPSTONE Project/Internship	0	0	32	32	16
		Total			32	32	16

SEMESTER - I

Dr D. Y. Patil School of Science &Technology,
Dr D. Y. Patil Vidyapeeth, Pimpri, Pune
MCA-CA-101: Mathematical Foundations of Computer Science

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	**04**	**Internals(TH): 40 Marks** **End_Semester(TH): 30 Marks**

Prerequisites: Basic knowledge of fundamental mathematics is required.

Course Objectives:

The students will be better learning the concepts of graph, matrix etc.

Course Outcomes:

Upon successful completion of this course, students will be able to:

The students will able to:

CO1. Inculcate critical thinking to carry out scientific investigation objectively without being biased with preconceived notions.

CO2. Equip the student with skills to analyze problems, formulate a hypothesis, evaluate and validate results, and draw reasonable conclusions thereof.

CO3. Understand sets, relations, functions, and discrete structures

CO4. Prepare students for pursuing research or careers in industry in mathematical sciences and allied fields.

CO5. Imbibe effective scientific and/or technical communication in both oral and writing.

CO6. Able to model and solve real-world problems using Gauss Jordan Methods

Course Contents

Unit I	**Introduction**	**(10 Hours)**
Simple and multi graphs, directed and undirected graphs, Eulerian and Hamiltonian Graphs, Shortest path algorithms, Chromatic number, Bipartite graph, graph coloring.		
***Mapping of Course Outcomes**	CO1	
Unit II	**Sets and Relations**	**(09 Hours)**
Definition of sets, subsets, complement of a set, universal set, intersection and union of sets, De-Morgan's laws, Cartesian products, Equivalent sets, Countable and uncountable sets, min-set, Partitions of sets, Relations: Basic definitions, graphs of relations, properties of relations		
***Mapping of Course Outcomes**	CO2,CO3	
Unit III	**Algebra of logic**	**(08 Hours)**

Propositions, Connectives, Tautologies and contradiction, Equivalence and implication, Principle of Mathematical induction, quantifiers.		
*Mapping of Course Outcomes	CO4	
Unit IV	**Introduction of a Matrix**	**(10 Hours)**
Its different kinds, matrix addition and scalar multiplication, multiplication of matrices, transpose etc. Square matrices, inverse and rank of a square matrix, solving simultaneous equations using Gauss elimination, Gauss Jordan Methods, Matrix Inversion method.		
*Mapping of Course Outcomes	CO5,CO6	

Learning Resources

Reference Books:

1. Discrete Mathematical structure for Computer Sciences Publications. Kolman and Busby PHI
2. Discrete Mathematical Structures for Computer Science B Kolman& R.C McGraw-Hill

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	1	-	-	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-	-	-
CO4	1	2	-	2	-	-	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-	-	-

Dr D. Y. Patil School of Science &Technology, Dr D. Y. Patil Vidyapeeth, Pimpri, Pune

MCA-CA-102: OOP using Java

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	**05**	**Mid_Semester(TH): 40 Marks**
		End_Semester(TH): 60 Marks

Prerequisites: Basic knowledge of Java syntax, as well as basic Java programming concepts.

Course Objectives:

1. To understand the basic concepts and fundamentals of platform independent object-oriented language.
2. To demonstrate skills in writing programs using exception handling techniques and multithreading.
3. To understand streams and efficient user interface design techniques.

Course Outcomes:

Upon successful completion of this course, Students will be able to:

CO1. Use the syntax and semantics of java programming language and basic concepts of OOP.

CO2. Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.

CO3. Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.

CO4. Design event driven GUI and web related applications which mimic the real word scenarios.

CO5. Implement Program. Write code. Perform unit testing. Integrate subsystems. Resolve defects and revise and adapt existing code.

CO6. Test and Validate Program. Develop test procedures. Perform tests.

Course Contents

Unit I	**Introduction**	**(10 Hours)**
Need for OOP Paradigm, Summary of OOP Concepts, Coping with Complexity, Abstraction Mechanisms, A Way of Viewing World – Agents, Responsibility, Messages, Methods, History of Java, Java Buzzwords, Data Types, Variables, Scope and Life Time of Variables, Arrays, Operators, Expressions, Control Statements, Type Conversion and Casting, Simple Java Program, Concepts of		

Classes, Objects, Constructors, Methods, Access Control, This Keyword, Garbage Collection, Overloading Methods and Constructors, Method Binding, Inheritance, Overriding and Exceptions, Parameter Passing, Recursion, Nested and Inner Classes, Exploring String Class.		
***Mapping of Course Outcomes**	CO1	
Unit II	**Inheritance, Packages and Interfaces**	**(09 Hours)**
Inheritance, Packages and Interfaces: Hierarchical Abstractions, Base Class Object, Subclass, Subtype, Substitutability, Forms of Inheritance-Specialization, Specification, Construction, Extension, Limitation, Combination, Benefits of Inheritance, Costs of Inheritance. Member Access Rules, Super Uses, Using Final with Inheritance, Polymorphism-Method Overriding, Abstract Classes, The Object Class. Defining, Creating and Accessing a Package, Understanding Class path, Importing Packages, Differences between Classes and Interfaces, Defining an Interface, Implementing Interface, Applying Interfaces, Variables in Interface and Extending Interfaces, Exploring Java.IO.		
***Mapping of Course Outcomes**	CO2	
Unit III	**Exception Handling and Multithreading**	**(07 Hours)**
Exception Handling and Multithreading: Concepts of Exception Handling, Benefits of Exception Handling, Termination or Presumptive Models, Exception Hierarchy, Usage of Try, Catch, Throw, Throws and Finally, Built in Exceptions, Creating Own Exception Sub Classes. String Handling, Exploring Java.Util, Differences between Multi-Threading and Multitasking, Thread Life Cycle, Creating Threads, Thread Priorities, Synchronizing Threads, Interthread Communication, Thread Groups, Daemon Threads. Enumerations, Autoboxing, Annotations, Generics.		
***Mapping of Course Outcomes**	CO3	
Unit IV	**Event Handling**	**(07 Hours)**
Event Handling: Events, Event Sources, Event Classes, Event Listeners, Delegation Event Model, Handling Mouse and Keyboard Events, Adapter Classes.		

The AWT Class Hierarchy, User Interface Components- Labels, Button, Canvas, Scrollbars, Text Components, Check Box, Check Box Groups, Choices, Lists Panels – Scrollpane, Dialogs, Menu bar, Graphics, Layout Manager – Layout Manager Types – Border, Grid, Flow, Card and Grid Bag.

***Mapping of Course Outcomes**	**CO4,CO6**

Unit V	**Applets**	**(07 Hours)**

Applets:

Concepts of Applets, Differences between Applets and Applications, Life Cycle of an Applet, Types of Applets, Creating Applets, Passing Parameters to Applets.

Swing:

Introduction, Limitations of AWT, MVC Architecture, Components, Containers, Exploring Swing- Japplet, Jframe and Jcomponent, Icons and Labels, Text Fields, Buttons – The Jbutton Class, Check Boxes, Radio Buttons, Combo Boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

***Mapping of Course Outcomes**	**CO5**

Practical List:

1. Write a java program to find the Fibonacci series using recursive and non-recursive functions
2. Write a java program to multiply two given matrices
3. Write a java program for Method overloading and Constructor overloading
4. Write a java program to display the employee details using Scanner class
5. Write a java program that checks whether a given string is palindrome or not
6. Write a java program to represent Abstract class with example. Write a java program to implement Interface using extends keyword.
7. Write a java program to create inner classes. Write a java program to create user defined package
8. Write a java program for creating multiple catch blocks. Write a java program for producer and consumer problem using Threads
9. Write a Java program that implements a multi-thread application that has three threads.
10. Write a java program to display File class properties. Write a java program to represent the Array List class. Write a Java program loads phone no, name from a text file using hash table
11. Write an applet program that displays a simple message
12. Write a Java program compute factorial value using Applet. Write a program for passing parameters using Applet

Learning Resources

Text Books:

1. Java the Complete Reference, 7th Edition, Herbert Schildt, TMH.
2. Understanding OOP with Java Updated Edition, T. Budd, Pearson Education.

Reference Books:

1. An Introduction to Programming and OO Design using Java, J. Nino and F.A. Hosch, John Wiley & Sons.
2. An Introduction to OOP, Third Edition, T. Budd, Pearson Education.
3. Introduction to Java Programming, Y. Daniel Liang, Pearson Education.
4. An Introduction to Java Programming and Object-Oriented Application Development, R.A. Johnson-Thomson.
5. Core Java 2, Vol 1, Fundamentals, Cay. S. Horstmann and Gary Cornell, Eighth Edition, Pearson Education.
6. Core Java 2, Vol 2, Advanced Features, Cay. S. Horstmann and Gary Cornell, eighth Edition, Pearson Education

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	1	2	1	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-
CO4	1	2	-	2	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-

Dr D. Y. Patil School of Science &Technology,
Dr D. Y. Patil Vidyapeeth, Pimpri, Pune
MCA-CA-103: Data Structures & Algorithms

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	**05**	**Mid_Semester(TH): 40 Marks**
		End_Semester(TH): 60 Marks

Prerequisites: Students must have knowledge of programming language, basics of mathematics and ability to write algorithms. Students also must have a good command on C & Python Programming.

Course Objectives:

1. To understand the basic concepts in data structure.
2. To discuss various algorithmic strategies to solve real life problems.
3. To acquaint the learner various data searching and sorting techniques.
4. To identify and use the appropriate data structure for various real-life problems using computer languages.
5. To understand the concepts of linear, non-linear data structures with its complexities.
6. To understand and efficiently apply various data structures such as stacks, queues, linked lists, trees and graphs for solving various computing problems using Python programming language.

Course Outcomes:

Upon successful completion of this course.

Students will be able to:

CO1. To understand the need of data structures.

CO2. To learn to apply the algorithm complexity techniques for various estimations.

CO3. To use organized data structure to solve various problem statements.

CO4. To develop the solutions to social issues using NP Complete theory using Python.

CO5. To distinguish the use of various structures in solving problems.

CO6. To understand the usage of appropriate data structures to implement algorithms.

Course Contents

Unit I	**Introduction to Data Structure and Algorithms**	**(07 Hours)**
Algorithm characteristics, Algorithm design tools, pseudo code and flowchart, Asymptotic notations complexity Recursion and iteration, recurrence equation, Master's theorem recurrence relationships. Need of Data Structure, Types of Data Structure and Abstract Data types.		
***Mapping of Course Outcomes**	CO1	
Unit II	**Linear Data Structures**	**(08 Hours)**
Arrays based Linear Data Structure: Array storage, sparse arrays; Transpose, addition, and multiplication of sparse matrices, Stacks and Queues and their applications, multiple stacks, queues in an array.		
***Mapping of Course Outcomes**	CO2	
Unit III	**Non-Linear Data Structures**	**(08 Hours)**
Singly, Doubly & Circular Linked Lists; representation, operations, applications, linked stacks and queues. linked lists based polynomial addition		
***Mapping of Course Outcomes**	CO3	
Unit IV	**Advanced Data Structures**	**(07 Hours)**
Event Handling: Trees, Basic concepts and definitions of a tree and binary tree and associated terminology, Binary tree traversal techniques, some more operations on binary trees, Heaps, heapsort.		
***Mapping of Course Outcomes**	CO4	
Unit V	**Searching & Sorting Techniques**	**(08 Hours)**
Searching techniques: Linear and Binary Search techniques, Sorting techniques: Insertion, Selection, Bubble, Merge sort, Quicksort.		
***Mapping of Course Outcomes**	CO5	
Unit VI	**NP–Hard and NP Complete Problems**	**(08 Hours)**
Definitions, Cook's Theorem, NP complete Problems, NP Hard Scheduling problems, Case studies		

***Mapping of Course Outcomes**	CO6

List of Practical:

1. Write Python programs for implementing the following searching techniques. a. Linear search b. Binary search c. Fibonacci search

2. Write Python programs for implementing the following sorting techniques to arrange a list of integers in ascending order. a. Bubble sort b. Insertion sort c. Selection sort

3. Write Python programs for implementing the following sorting techniques to arrange a list of integers in ascending order. a. Quick sort b. Merge sort

4. Write Python programs to a. Design and implement Stack and its operations using List. b. Design and implement Queue and its operations using List.

5. Write Python programs for the following: a. Uses Stack operations to convert infix expression into postfix expression. b. Uses Stack operations for evaluating the postfix expression

6. Write Python programs for the following operations on Single Linked List. (i) Creation (ii) insertion (iii) deletion (iv) traversal b. To store a polynomial expression in memory using single linked list

7. Write Python programs for the following operations on Circular Linked List. (i) Creation (ii) insertion (iii) deletion (iv) traversal

8. Write Python programs for the following: Uses functions to perform the following operations on Double Linked List. (i) Creation (ii) insertion (iii) deletion (iv) traversal in both ways

9. Write a Python program to implement Stack using linked list.

10. Write a Python program to implement Linear Queue using linked list

11. Write Python programs to implement the following graph traversal algorithms: a. Depth first search. b. Breadth first search

12. Write a Python program to perform the following: a. Create a binary search tree. b. Traverse the above binary search tree recursively in pre-order, post-order and in-order. c. Count the number of nodes in the binary search tree

Learning Resources

Reference Books:

1. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.

2. R.L. Kruse: Data Structures & Program Design in C, PHI.

3. D.F. Knuth: The art of Computer Programming Vol 1, Narosa Publications, 1985.

4. Byron S. Gottfried & J K Chhabra: Theory and Problems of Programming with C Language, Schaum's Outlines Series, TMH, 2005.

5. David Griffiths, Introduction to Electrodynamics, 3rd edition, 1999, Prentice Hall

6. David Griffiths, Introduction to Quantum Mechanics, 2nd edition, 2005, Prentice Hall

7. Y Daniel Liang, "Introduction to Programming using Python", Pearson.

8. Benjamin Baka, David Julian, "Python Data Structures and Algorithms", Packt Publishers,2017.

9. Rance D. Necaise, "Data Structures and Algorithms using Python", Wiley Student Edition.

10. Martin Jones, "Python for Complete Beginners", 2015.

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	1	-	-	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-	-	-
CO4	1	2	-	2	-	-	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-	-	-

SEMESTER - II

Dr D. Y. Patil School of Science &Technology, Dr D. Y. Patil Vidyapeeth, Pimpri, Pune

MCA-CA-201: Advanced Java

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	**05**	**Mid_Semester(TH): 40** **End_Semester(TH): 60**

Prerequisites: Students must have knowledge of programming language, basics of mathematics and ability to solve problem. Students also must have a good command on Java

Course Objectives:

1. To understand different types of server-side programming and technologies like Servlets, JSP, ASP, EJB, JSF, PHP, Node.

2. Understand the various server-side Spring Frameworks, REST, SOAP, ORM, Security

Course Outcomes:

Upon successful completion of this course, students will be able to:

CO1. Understand advanced server-side programming concepts and use technologies like Servlets, JSP, JSF and ASP

CO2. Adopt conveniently, ORM technique to bridge object and relational models of data.

CO3. Develop, real world API and Services using SOAP and REST.

CO4. Create application using Node.js and JMS API that provides the facility to create, send and read messages.

CO5. Efficiently create fast, secure, and responsive web applications using Spring Framework.

CO6. Persisting business objects using JPA, JMS and MySQL

Course Contents

Unit I	**Servlets, JSP, JSF and ASP**	**(06 Hours)**
JSP, JSTL, Spring Tag Libraries, Spring Controllers, Template & Layout, Spring Form Validations (Standard and Custom), jQuery, CSS3, Web Descriptor Language, AJAX, Web Socker Support, Java server Faces, JSF flows, UI Model-Framework – JSP, JSTL, Tiles/Thymeleaf, Spring MVC on Spring Boot, Hibernate Validato		
***Mapping of Course**	**CO1**	

Outcomes		
Unit II	**REST**	**(08 Hours)**
Webservices, Types of Webservices, REST, JAX-RS, Rest Frameworks, Rest Methods and APIs, REST Clients.		
*Mapping of Course Outcomes	**CO2**	
Unit III	**SOAP**	**(08 Hours)**
SOAP, JAX-WS, WSDL, SOAP Registries, SOAP Frameworks, SOAP Clients, Develop SOAP and REST API and Services. Framework – Spring MVC, Web-Services, Spring Security		
*Mapping of Course Outcomes	**CO3**	
Unit IV	**ORM**	**(07 Hours)**
Object Relation Mapping, JPA, Hibernate, Entity – Annotations, Association and In heritance mapping, Hibernate Session and Transaction, Caching, Native Query, HQL, Batch Processing and Intercepting Filter, Criteria Builder, Projections API, Named & Native Query. Framework – Spring Data JPA, Hibernate and JPA,MySQL/any rdbms Database		
*Mapping of Course Outcomes	**CO4**	
Unit V	**JMS, Node JS**	**(08 Hours)**
JMS, Queues and Topics, Creating Queues and Topics, Sending and Receiving messages using Queues and Topics. Introduction to Node JS, Benefits and Features, NPM in Node JS, Event Handling. Framework – ActiveMQ or RabbitMQ, Spring JMS integration, NodeJS, NPM		
*Mapping of Course Outcomes	**CO5**	
Unit VI	**Spring Framework**	**(08 Hours)**
Developing a Batch Application that gets executed in the background process, and gets triggered at a specific regular interval, Task/Tasklet, Steps, Sharing Batch Context Information between Steps Exception Handling, Transaction Commit Intervals, Chunk Processing, File/DB/JMS based Reader and Writers. Framework – Spring Boot, Spring Batch, Spring Data JPA, JMS and MySQL		
*Mapping of Course Outcomes	**CO5, CO6**	

List of Practical:

1. Develop a web application with AJAX and UI model framework 5 hours

2. Create an application implementing a RESTful API 5 hours

3. Create Web application using HTML, CSS and Node.js 5 hours

4. Integrate Spring with ORM framework 5 hours

5. Develop Web Applications using Spring Framework 5 hours

6. Create UI Management for Spring Boot and Node js applications 5 hours

Learning Resources

Reference Books:

1. Christian Bauer, Gavin King, Gary Gregory, Linda Demichiel, Java Persistence with Hibernate, 2ed, MANNING Publications, 2016

2. David R. Heffelfinger, Java EE 8 Application Development, Packt Publishing, 2017.

3. Dhruti Shah, Node .js Guidebook, First edition, BPB Publications, 2018.

4. https://microservices.io/

5. https://javaee.github.io/javaee-spec/

6. https://spring.io/projects/ 6. https://nodejs.org/en/

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	1	-	-	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-	-	-
CO4	1	2	-	2	-	-	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-	-	-

Dr D. Y. Patil School of Science &Technology,
Dr D. Y. Patil Vidyapeeth, Pimpri, Pune
MCA-CA-202: Artificial Intelligence

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	04	Mid_Semester(TH): 40 End_Semester(TH): 60

Prerequisites: Students needs to have basic knowledge of linear algebra, vector, matrix, probability, Propositional Logic & python programming.

Course Objectives:

The objective of this course is:

1. To understand the concepts of Artificial Intelligence (AI).
2. To understand strength of and weakness of searching algorithms.
3. To learn and compare the searching techniques for AI applications.
4. To acquaint with the various knowledge representation & experts' systems.
5. To understand basic probability notations in artificial Intelligence/ Game theory.
6. To acquaint with the fundamentals of knowledge presentations and reasoning.

Course Outcomes:

After completion of the course, students should be able to

CO1. To understand the fundamentals of Artificial Intelligence

CO2. To design smart system using different search strategies of Artificial Intelligence

CO3. To analyze various basic probability notations, game theory

CO4. To apply various algorithms for Artificial Intelligence application development

CO5. To implement Artificial Intelligence solutions using logical reasoning

CO6. To analyze the knowledge presentation and expert systems

Course Contents

Unit I	Introduction	(07 Hours)
Introduction: History & overview of Artificial Intelligence, Different Definitions, Problem Solving Strategies, Applications, Physical Symbol System Hypothesis, production systems, Characteristics of production, Agents and Environments – Concept of rationality – Nature of environments – Structure of agents.		

*Mapping of Course Outcomes	CO1	
Unit II	Searching Techniques	(07 Hours)
Uninformed Search, depth first search, breadth first search, Heuristic Search Strategies (Greedy Best First Search, A* Search, Memory Bounded Heuristic Search) Evolutionary algorithms Local Search Algorithms (Hill-Climbing Search, Simulated Annealing Search, Local Beam Search)		
*Mapping of Course Outcomes	CO2	
Unit III	Basic Probability Notation	(08 Hours)
Inference Using Full Joint Distribution, Independence, Bayes' Rule and it's Use The Planning Problem, Planning with State Space Search, Planning Graphs, Efficient Representation of Conditional Distribution, Exact Inference, Approximate Inference Extending Probability to First Order Representations Alternatives for Uncertain Reasoning		
*Mapping of Course Outcomes	CO3	
Unit IV	Game Playing	(08 Hours)
Constraint Satisfaction Problems(CSP), constraint propagation, backtracking search for CSP, local search for CSP, structure of CSP , Minimax & Alpha-Beta Pruning Algorithm, Imperfect Real-time decisions, Knowledge Based Agents, Example, Propositional Logic, Reasoning Patterns in Propositional Logic, Syntax and semantics of First Order Logic, Inference in First Order Logic Knowledge Base Reasoning Systems for Categories (Semantic Networks, Description Logics), Reasoning with default Information Acting under uncertainty,		
*Mapping of Course Outcomes	CO4	
Unit V	Formalized & Propositional Logic	(06 Hours)
Formalized symbolic logic: Propositional logic-first order predicate logic, wff conversion to clausal form, inference rules, the resolution principle, Dealing with inconsistencies and uncertainties, fuzzy logic. Probabilistic Reasoning Structured knowledge, graphs, frames and related structures, Knowledge organization and manipulation. Matching Techniques, Knowledge organizations, Management.		
*Mapping of	CO5	

Course Outcomes		
Unit VI	**Knowledge Representation and Expert Systems**	**(08 Hours)**
Knowledge representation, Natural Language processing, Pattern recognition, expert systems, introduction to machine learning Case Study: Sentiment Analysis, Case Study: Object Recognition. Ontological engineering		
*Mapping of Course Outcomes	**CO6**	

PRACTICALS (any 9)

1. Study & list tuple, set, dictionary, classes, inheritance in Python
2. Study and understand simple reflex and Model Based Agent
3. Implement graph in Python for profit or loss in banking application
4. Describe the given problem statement using PEAS description.
5. Implement basic searching algorithm for given AI problem
6. Write a program to solve 8 Queens' problem
7. Implement memory bounded A* & A* algorithm for given problem.
8. Implement Alpha Beta Tree search.
9. Implement classical planning algorithm
10. Solve Robot Obstacle/transversal problem means end analysis.

Learning Resources

Reference Books:

Artificial Intelligence: Elaine Rich, Kevin Knight, Mc-GrawHill.

2. Introduction to AI & Expert System: Dan W. Patterson, PHI.

3. Artificial Intelligence by Luger (Pearson Education)

4. Artificial Intelligence, A Modern Approach. Stuart Russell and Peter Norvig.

5. Thomas Haslwanter, "An Introduction to Statistics with Python with Applications in the Life 6. Sciences", Springer International Publishing Switzerland 2016, ISBN 978-3-319- 28315-9, ISBN 978-3-319-28316-6 (eBook)

7. Peter Bruce and Andrew Bruce, "Practical Statistics for Data Scientists", First Edition,

O’Reilly Media, ISBN-978-1-491-95296-2

8. Allen B. Downey, "Think Stats", Second Edition, O’Reilly Media, ISBN: 978-1-491- 90733-7

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	1	-	-	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-	-	-
CO4	1	2	-	2	-	-	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-	-	-

Dr D. Y. Patil School of Science &Technology,
Dr D. Y. Patil Vidyapeeth, Pimpri, Pune
MCA-CA-203: Database Management Systems

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	**04**	**Mid_Semester(TH): 40** **End_Semester(TH): 60**

Prerequisites: Basic knowledge of Data Structures and Algorithms, Discrete Mathematics is required.

Course Objectives:

1. To understand the fundamental concepts and the applications of Database Management Systems.
2. To acquire the skillset to use flexible databases for real applications.
3. To get familiar with Data Collection and Design techniques.
4. To design a Database Management Systems for scalable projects.
5. To relate different DB languages like MySQL, Noe4J, Risk, MongoDB.
6. To understand the relational database design principles.

Course Outcomes:

Upon successful completion of this course, students will be able to:

CO1. To analyze and design the basic elements of a relational database management system.

CO2. To learn to normalize the databases using single value normalization.

CO3. To identify the relevant data models for problems.

CO4. To design and evaluate entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respect data into RDBMS and formulate SQL queries on the data.

CO5. To interpret the query evaluation and optimization techniques.

CO6. Apply Nosql development tools on different types of NoSQL Databases.

Unit I	**Introduction to Database**	**(06 Hours)**
Database Concepts, Database System Architecture and Data Modeling: Data Models, Basic Concepts, entity, attributes, relationships, constraints, keys. E-R and EER diagrams: Components of E-R Model, conventions, converting E-R diagram into tables, EER Model components, converting EER diagram into tables, legacy system model. Relational Model: Basic concepts, Attributes and		

Domains, Codd's Rules. Relational Integrity: Domain, Entity, Referential Integrities, Enterprise Constraints, Schema Diagram. Relational Algebra: Basic Operations, Selection, projection, joining, outer join, union, difference, intersection, Cartesian product, division operations (examples of queries in relational algebraic using symbols).		
***Mapping of Course Outcomes**	**CO1**	
Unit II	**Data Collection**	**(06 Hours)**
Data Processing - Data collection; Data preparation; Training a model on the data; Evaluation of the model performance; Data visualization techniques and inferences - scatter plot, scatter matrix, histogram, box plot.		
***Mapping of Course Outcomes**	**CO2**	
Unit III	**Database Design &SQL**	**(08 Hours)**
Functional Dependency, Purpose of Normalization, Data Redundancy and Update Anomalies, Single Valued Normalization: 1NF, 2NF, 3NF, BCNF. Decomposition: lossless join decomposition and dependency preservation, Multi valued Normalization (4NF), Join Dependencies and the Fifth Normal Form. Introduction to SQL: Characteristics and advantages, SQL Data Types and Literals, DDL, DML, DCL, SQL Operators, Tables: Creating, Modifying, Deleting, Views: Creating, Dropping, Updating using Views, Indexes, Nulls SQL DML Queries: SELECT Query and clauses, Set Operations, Predicates and Joins, set membership, Tuple Variables, set comparison, Ordering of Tuples, Aggregate Functions, Nested Queries, Database Modification using SQL Insert, Update and Delete Queries		
***Mapping of Course Outcomes**	**CO3**	
Unit IV	**Query Processing and Database transactions**	**(06 Hours)**
Query Processing: Overview, Measures of query cost, Evaluation of expression, Materialization and Pipelining algorithm. Transaction: Basic concept of a Transaction, Transaction Management, Properties of Transactions, Concept of Schedule, Serial Schedule, Serializability: Conflict and View, Cascaded Aborts, Recoverable and No recoverable Schedules. Concept of Stored Procedures, Cursors, Triggers, assertions, roles and privileges Programmatic SQL: Embedded SQL, Dynamic SQL, Advanced SQL-Programming in MYSQL.		
***Mapping of Course Outcomes**	**CO4**	
Unit V	**Concurrency Control**	**(07 Hours)**

Concurrency Control: Need, Locking Methods, Deadlocks, Time-stamping Methods, and Optimistic Techniques. Recovery Methods: Shadow-Paging and Log-Based Recovery, Checkpoints, Performance Tuning, Query Optimization		
***Mapping of Course Outcomes**	**CO5**	
Unit VI	**NoSQL databases**	**(07 Hours)**
Introduction, Overview, and History of NoSQL Databases – The Definition of the Four Types of NoSQL Databases, Column-oriented NoSQL databases using Apache HBASE, Column-oriented NoSQL databases using Apache Cassandra NoSQL Key/Value databases using MongoDB, NoSQL Key/Value databases using Riak, Graph NoSQL databases using Neo4J, NoSQL database development tools and programming languages Future Trends for NoSQL databases		
***Mapping of Course Outcomes**	**CO6**	

Practical List:

Assignments from all Groups (A, B, C) are compulsory.

Group- A:

1. Draw E-R diagram and convert entities and relationships to relation table for a given scenario. a. Two assignments shall be carried out i.e. consider two different scenarios (e.g. bank, college)

2. Install and configure client and server for MySQL and MongoDB (Show all commands and necessary steps for installation and configuration).

3. Perform the following: a. Viewing all databases, creating a Database, viewing all Tables in a Database, Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)

4. Perform the following: a. Altering a Table, Dropping/Truncating/Renaming Tables, backing up / Restoring a Database.

5. For a given set of relation schemes, create tables and perform the following Simple Queries, Simple Queries with Aggregate functions, Queries with Aggregate functions (group by and having clause), Queries involving- Date Functions, String Functions, Math Functions Join Queries- Inner Join, Outer Join Subqueries- with IN clause, With EXISTS clause.

6. For a given set of relation tables perform the following a. Creating Views (with and without check option), Dropping views, Selecting from a view.

7. Write a Pl/SQL program using FOR loop to insert ten rows into a database table.

8. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID) write a cursor to select the five highest paid employees from the table.

9. Illustrate how you can embed PL/SQL in a high-level host language such as C/Java and demonstrates how a banking debit transaction might be done.

10. Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.

11. Study the Riak database and its uses. Also elaborate on building and installing of Riak.

Group B-

MongoDB/Apache Cassandra Queries:

1. Design and Develop MongoDB/Apache Cassandra Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.).

2. MongoDB/Apache Cassandra - Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.

3. MongoDB/Apache Cassandra - Map reduces operations: Implement Map reduces operation with suitable example using MongoDB.

4. Database Connectivity: Write a program to implement MongoDB database connectivity

Note* - Teachers can take the flexibility to use any other advanced tools Instead of MongoDB/Apache Cassandra

Group C-

Using the database concepts covered in Group A and Group B, develop an application with following details: 1. Follow the same problem statement decided in Assignment -1 of Group A.

2. Follow the Software Development Life cycle and other concepts learnt in Software Engineering Course throughout the implementation.

3. Develop application considering:

- Front End: Java/Perl/PHP/Python/Ruby/.net/any other language
- Backend: MongoDB/MySQL/Oracle

4. Test and validate application using Manual/Automation testing.

5. Student should develop application in group of 2-3 students and submit the Project Report which will consist of documentation related to different phases of Software Development Life Cycle:

- Title of the Project, Abstract, Introduction
- Software Requirement Specification
- Conceptual Design using ER features, Relational Model in appropriate Normalize form
- Graphical User Interface, Source Code
- Testing document
- Conclusion.

Learning Resources

Test Books:

1. Raghurama Krishnan, Johannes Gehrke, Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India

2. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India.

Reference Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2005), Database System Concepts, 5th edition, McGraw-Hill, New Delhi, India.

2. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7thedition

@The CO-PO Mapping Matrix

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	1	-	-	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-	-	-
CO4	1	2	-	2	-	-	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-	-	-