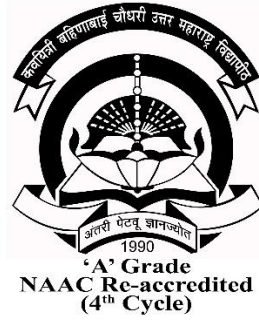


**KAVAYITRI BAHINABAI CHAUDHARI  
NORTH MAHARASHTRA UNIVERSITY, JALGAON**

॥अंतरी पेटवू ज्ञानज्योत॥



**MASTER OF COMPUTER APPLICATIONS  
(MCA)**

**(at University Campus under Academic Flexibility w.e.f. 2024-25)**

**(w.e.f. Academic Year 2024-25)**

**KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON**  
**School of Computer Sciences, M.C. A. (Computer Application) PROGRAMME (A. Y. 2024-25)**  
**Credit distribution structure for Two Years/ Programme**

Year (2 Yr PG)	Sem	Major (Core) Subjects		RM	OJT/FP	RP	Cumulative Credits	Degree
		Mandatory(MCS)	Elective(MCSE)					
<b>I (Level 6.0)</b>	<b>Sem-I</b>	MCA-411: Database Management System (DBMS) (4) (T)  MCA-412: Operating Systems (4) (T)  MCA- 413: Fundamentals of Artificial Intelligence (4)(T)  MCA-414: Object Oriented Programming Using C++ (4)(T)  MCA-415: Lab on DBMS (2)(P)  MCA-417: Lab on OS (Linux) (2) (P)  MCA-418: Lab on C++ Programming (2) (P)	MCA-419 (A): Computer Programming and Problem Solving (CPPS) (2)(T)  MCA-419 (B): Lab on CPPS(2)(P)  OR  MCA-420 (A): Web Programming (2)(P)  MCA-420 (B): Lab on Web Programming (2)(P)	-	-	-	26	<b>PG Diploma (After 1-Yr PG Degree)</b>
	<b>Sem-II</b>	MCA-421: Advanced Software Development Methodologies (4)(T)  MCA-422 Data Structures and Algorithms(4)(T)  MCA-423 Java Programming (Core Java) (2)(T)  MCA-424: Lab on Data Structures and Algorithms (2)(P)	MCA-426 (A): Digital Image Processing (DIP) (2) (T)  MCA-426 (B): Lab on DIP (2) (P)  OR MCA-427 (A); Python Programming(2)(T)  MCA-427 (B): Lab on Python Programming (2) (P)	MCA-416: Research Methodology (4)(T)	-	-	26	

		MCA-425: Lab on Java Programming(2)(P)	MCA-428 (A): Machine Learning(2)(T)  MCA-428 (B): Lab on Machine Learning (2) (P)  <b>OR</b> MCA-429 (A): Data Science (2)(T)  MCA-429 (B): Lab on Data Science (2) (P)					
<b>Exit Option: PG Diploma</b> after ONE Year PG Degree (with additional 4 Credits of OJT)								
<b>II (Level 6.5)</b>	<b>Sem-III</b>	MCA-511: Design and Analysis of Algorithms (4)(T)  MCA-512: Advanced Java Programming (2)(T)  MCA-513: Lab on Advanced Java Programming(2)(P)	MCA-514 (A): Natural Language Processing (2) (T) CA-514 (B): Lab on Natural Language Processing (2) (P)  <b>OR</b> MCA-515 (A): AI in Practice with Python (2) (T) MCA-515(B): Lab on AI in Practice with Python (2) (P)  <b>OR</b> CA-516 (A): High Performance Computing Paradigms and Applications (2) (T) CA-516 (B): Lab on High Performance Computing (2) (P)	-	-	MCA-520: Research Project (6)(P)	22	<b>PG Degree MCA After 2-Yr PG</b>
			CA-517 (A): Mobile Application Development (Android Programming) (2) (T) CA-517 (B): Lab on Mobile Application Development (Android Programming) (2) (P)  <b>OR</b> CA-518 (A): Microsoft .Net Technologies (4) (T) CA-518 (B): Lab on Microsoft .Net Technologies (4) (T)					

			<b>OR</b> CA-519 (A): Ruby on Rails (2) (T)  CA-519 (B): Lab on Ruby on Rails (2) (P)					
	<b>Sem-IV</b>	-	MCA-521: 4 Credits (T) MCA-522: 4 Credits (T)  <b>(NOTE:</b> TWO available NPTEL Swayam Courses will be selected in November/December for Registration)	-	MCA-523 On Job Training (OJT) (12) (P)	-	<b>20</b>	
<b>Cumulative Credits for 1 Year PG Degree</b>		<b>36</b>	<b>12</b>	<b>4</b>	-	-	<b>52</b>	
<b>Cumulative Credits for 2 Year PG Degree</b>		<b>44</b>	<b>28</b>	<b>4</b>	<b>12</b>	<b>6</b>	<b>94</b>	

**Abbreviations:** Yr.: Year; Sem.: Semester; OJT: On Job Training; Internship/ Apprenticeship; FP: Field projects; RM: Research Methodology; Research Project: RP; Cumulative Credits: Cum. Cr., T- Theory Course, P – Practical course, MMCA- School Specific Core Course, MCSE- School Specific Elective Course

**Note:** The courses which do not have practical, 'P' will be treated as 'T'.

**(NOTE:** For MCA-521: 4 Credits (T), MCA-522: 4 Credits (T), TWO available NPTEL Swayam Courses will be selected in November/December for Registration on the NPTEL Portal, The Academic Committee will select/finalize the suitable courses (particularly Coding/Programming/Software Skills courses) and the Swayam Coordinator of the school and the Admission Committee will initiate the registration process on the NPTEL portal in November/December of the A.Y.)

**KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON**  
**School of Computer Sciences, PG DEGREE M. C. A. (Computer Application) PROGRAMME (A. Y. 2024-25)**  
**Credit distribution structure for Two Years M.C. A. Degree Programme**

**Teaching and Examination scheme, Master of Computer Application (M. C. A.)**  
**M.C.A. (Level 6.0) Sem- I (Name of Courses for - Major, RM, OJT, RP courses)**

Sr No	Course Category	Name of the course(Title of the Paper)		Total Credit	Hours/ Semester	Teaching Scheme (hrs/week)		Evaluation Scheme		
						Theory	Practical	Continuous Internal Evaluation (CIE) (CA)	End Semester Evaluation (ESE) (UA)	Duration of Examination (Hrs)
						T	P			
1	Major	MCA-411	Database Management System (DBMS)	4	60	4	-	40	60	3
		MCA-412	Operating Systems	4	60	4	-	40	60	3
		MCA- 413	Fundamentals of Artificial Intelligence	4	60	4	-	40	60	3
		MCA-414	Object Oriented Programming Using C++	4	60	4	-	40	60	3
		MCA-415	Lab on DBMS	2	60	-	4	20	30	2
		MCA-417	Lab on OS (Linux)	2	60	-	4	20	30	2
		MCA-418	Lab on C++ Programming	2	60	-	4	20	30	2
2	Elective (Any One Group)	MCA-419 (A)	Computer Programming and Problem Solving (CPPS)	2	30	2	-	20	30	2
		MCA-419 (B)	Lab on CPPS	2	60	-	4	20	30	2
		OR								
		MCA-420 (A)	Web Programming	2	30	2	-	20	30	2
		MCA-420 (B)	Lab on Web Programming	2	60	-	4	20	30	2

Total	26	510	18	16	260	390	-
-------	----	-----	----	----	-----	-----	---

**KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON**  
**School of Computer Sciences, PG DEGREE M. C. A. (Computer Application) PROGRAMME (A. Y. 2024-25)**  
**Credit distribution structure for Two Years M.C. A. Degree Programme**

**Teaching and Examination scheme, Master of Computer Application (M. C. A.)**  
**M.C.A. (Level 6.0) Sem- II (Name of Courses for - Major, RM, OJT, RP courses)**

Sr No	Course Category	Name of the course(Title of the Paper)		Total Credit	Hours/ Semester	Teaching Scheme (Hrs/Week)		Evaluation Scheme		
						Theory	Practical	Continuous Internal Evaluation (CIE) (CA)	End Semester Evaluation (ESE) (UA)	Duration of Examination (Hrs)
						T	P			
1	Major	MCA-421	Advanced Software Development Methodologies	4	60	4	-	40	60	3
		MCA-422	Data Structures and Algorithms	4	60	4	-	40	60	3
		MCA-423	Java Programming (Core Java)	2	30	2	-	20	30	2
		MCA-424	Lab on Data Structures and Algorithms	2	60	-	4	20	30	2
		MCA-425	Lab on Java Programming	2	60	-	4	20	30	2
2	Elective 1 (Select Any One Group)	MCA-426 (A)	Digital Image Processing (DIP)	2	30	2	-	20	30	2
		MCA-426 (B)	Lab on DIP	2	60	-	4	20	30	2
		OR								
		MCA-427(A)	Python Programming	2	30	2	-	20	30	2
		MCA-427(B)	Lab on Python Programming	2	60	-	4	20	30	2
	Elective 2 (Select Any One Group)	MCA-428 (A)	Data Science	2	30	2	-	20	30	2
		MCA-428 (B)	Lab on Data Science	2	60	-	4	20	30	2
		OR								
		MCA-429 (A)	Machine Learning	2	30	2	-	20	30	2
		MCA-429 (B)	Lab on Machine Learning	2	60	-	4	20	30	2
3	Research	MCA-416	Research Methodology	4	60	4	-	40	60	3
Total				26	510	18	16	260	390	-

**KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON**  
**School of Computer Sciences, PG DEGREE M. C. A. (Computer Application) PROGRAMME (A. Y. 2024-25)**  
**Credit distribution structure for Two Years M.C. A. Degree Programme**

**Teaching and Examination scheme, Master of Computer Application (M. C. A.)**  
**M.C.A. (Level 6.5) Sem- III (Name of Courses for - Major, RM, OJT, RP courses)**

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/ Semester	Teaching Scheme (Hrs/Week)		Evaluation Scheme		
						Theory	Practical	Continuous Internal Evaluation (CIE) (CA)	End Semester Evaluation (ESE) (UA)	Duration of Examination (Hrs)
						T	P			
1	Major	MCA-511	Design and Analysis of Algorithms	4	60	4	-	40	60	3
		MCA-512	Advanced Java Programming	2	30	2	-	20	30	2
		MCA-513	Lab on Advanced Java Programming	2	60	-	4	20	30	2
2	Elective-1	MCA-514 (A)	Natural Language Processing	2	30	2	-	20	30	2
		MCA-514 (B)	Lab on Natural Language Processing	2	60	-	4	20	30	2
		OR								
		MCA-515 (A)	AI in Practice with Python	2	30	2	-	20	30	2
		MCA-515 (B)	Lab on AI in Practice with Python	2	60	-	4	20	30	2
		OR								
		MCA-516 (A)	High Performance Computing Paradigms and Applications	2	30	2	-	20	30	2
		MCA-516 (B)	Lab on High Performance Computing	2	60	-	4	20	30	2
	Elective-2	MCA-517 (A)	Mobile Application Development (Android Programming)	2	30	2	-	20	30	2
		MCA-517 (B)	Lab on Mobile Application Development (Android Programming)	2	60	-	4	20	30	2
		OR								
		MCA-518 (A)	Microsoft .Net Technologies	2	30	2	-	20	30	2



		MCA-518 (B)	Lab on Microsoft .Net Technologies	2	60	-	4	20	30	2
		OR								
		MCA-519 (A)	Ruby on Rails	2	30	2	-	20	30	2
		MCA-519 (B)	Lab on Ruby on Rails	2	60	-	4	20	30	2
	Research Project	MCA-520	Research Project	6	180	-	6	50	100	3
Total				22	510	10	18	210	340	-

**KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON**  
**School of Computer Sciences, PG DEGREE M. C. A. (Computer Application) PROGRAMME (A. Y. 20243-25)**  
**Credit distribution structure for Two Years M.C. A. Degree Programme**

**Teaching and Examination scheme, Master of Computer Application (M. C. A.)**  
**M.C.A. (Level 6.5) Sem- IV (Name of Courses for - Major, RM, OJT, RP courses)**

Sr No	Course Category	Name of the course (Title of the Paper)		Total Credit	Hours/ Semester	Teaching Scheme (Hrs/Week)		Evaluation Scheme		
						Theory	Practical	Continuous Internal Evaluation (CIE) (CA)	End Semester Evaluatio n (ESE) (UA)	Duration of Examination (Hrs)
						T	P			
1	Major (NPTEL Swayam)	MCA-521	(NPTEL, Swayam) Course 1	4	60	4	-	-	100	3
		MCA-522	(NPTEL, Swayam) Course 2	4	60	4	-	-	100	3
2	FP/OJT, RP	MCA-523	On Job Training (OJT)	12	540	-	24	100	200	3
Total				20	660	8	30	100	400	--

**Note:** Major Courses MCA-521 and MCA-522 must be completed from NPTEL (Swayam). TWO available NPTEL Swayam Courses (Four Credit/Three Credit) will be selected in November/December for Registration. If the required Four Credit courses are not available, then the Three Credit courses may be selected. Then the marks obtained in Three Credit (NPTEL) courses will be converted to equivalent Four Credit courses marks.

# Semester-I

---

<i>Course Code: MCA-411</i>	<b>Database Management System (DBMS)</b>	<i>Clock Hours: 60</i> <i>Total Marks: 100</i> <i>Total Credits:04</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"><li>1) Introduction to the basic concepts of database management systems, learning to design databases using ER modelling, and decomposing data based on functional dependencies.</li><li>2) Understand Relational databases, SQL, Transaction management, Query processing, concurrency control and recovery system.</li><li>3) Describe and discuss selected advanced database topics such as distributed database and XML and Web data.</li></ol>		
<b>Unit-I</b>	<b>[05]</b>	<b>Max Marks:08</b>
<b>Introduction:</b> Database system application and purpose, Characteristics of DBMS, Database Users, 1-tier, 2-tier and 3-tier architecture of DBMS along with its advantages, Levels of Database Architecture, Data Models, Data-schemas and instances, Data Independence, Role and responsibilities of DBA.		
<b>Unit-II</b>	<b>[10]</b>	<b>Max Marks:12</b>
<b>Database Design and E-R Model:</b> Overviews of Database Design, ER Modelling concepts, ER Diagrams, Reduction to Relational Schemas, Extended ER Features, Alternative notations for Modelling, Cardinality constraints, Atomic Domains and 1NF, Decomposition using Functional Dependencies (BCNF, 3NF and 4NF).		
<b>Unit-III</b>	<b>[12]</b>	<b>Max Marks:20</b>
<b>Relational Databases:</b> Structure of Relational Databases, Database Schemas, Keys, Schema diagrams, Relational Query Languages, Relational Operation. Overview of SQL, SQL Data Definition, Basic Structure of SQL Queries, Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub queries, Modification of Databases. Join Expressions, Views, Transactions, Integrity Constraints, SQL data types and Schemas, Authorization, Accessing SQL from Programming Languages, Overview of Dynamic SQL and SQL CLI. Functions and Procedures, Triggers. The relational Algebra fundamental and extended Operations. Tuple and Domain Relational Calculus.		
<b>Unit-IV</b>	<b>[10]</b>	<b>Max Marks:22</b>
<b>Transaction Management and Query Processing:</b> Transaction Concept, Model, Storage Structure, Atomicity and Durability, Isolation, Levels of Isolation, Overview of Query Processing, Measuring Query Cost, Selection Operation, Sorting, Join Operation, Other Operations and Evaluation of Expression. Overview of Query Optimization, Transformation of Relational Expression, Choice of Evaluation Plan.		

<b>Unit-V</b>	<b>[10]</b>	<b>Max Marks:16</b>
<b>Concurrency Control and Recovery System:</b> Lock based Protocol, Timestamp based Protocol, Validation based Protocol, Deadlock Handling, Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithms, Buffer Management, Early lock release and logical undo operations, Remote Backup Systems. Case study: ARIES		
<b>Unit-VI</b>	<b>[13]</b>	<b>Max Marks:22</b>
<b>Advanced Topics in Databases:</b> Introduction to Object Databases: Shortcomings of Relational Data Model, The Conceptual Object Data Model, Objects in SQL:1999 and SQL:2003. Introduction to XML and Web Data: Semi-structured Data, Overview of XML, XML Data Definitions, XML Schema, XML Data Manipulation: XQuery, XPath Query Languages: XPath and SQL/XML.		
<b>Distributed Databases:</b> Overview, Homogeneous and Heterogeneous Databases, Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control, Cloud based Databases.		
<b>References:</b> <ol style="list-style-type: none"> <li>1) Michael Kifer, Arthur Bernstein, P.M, Lewis and P.K. Panigrahi (2011), “Database Systems: An Application Oriented Approach”, Second Edition, Pearson Education, 2011, ISBN: 9788131703748.</li> <li>2) C. J .Date, A. Kannan and S. Swamynathan (2006), “An Introduction to Database Systems”,</li> <li>3) Eighth Edition, Pearson Education, 2006, ISBN:978-81-7758-556-8</li> <li>4) Silberschatz, H.F.Korth, and S.Sudarshan (2011), “Database System Concepts”, TMH Publications, Sixth Edition, 2011, ISBN: 978-007-132522-6.</li> <li>5) Ramez Elmasri, Shamkant B. Navathe (2011), “Fundamentals of Database Systems” Seventh Edition, Pearson Education, 2011, ISBN: 978-0-13-397077-7.</li> </ol>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"> <li>1. Apply the relational model, specify integrity constraints, and explain how to create a relational database using an ER diagram and normalization techniques.</li> <li>2. Apply SQL to create, query and manipulate relational databases.</li> <li>3. Determine partitioning and distribution of data across networked nodes of a DBMS and data optimization in a distributed environment.</li> </ol>		

<i>Course Code: MCA-412</i>	<b>Operating Systems</b>	<i>Clock Hours: 60</i> <i>Total Marks: 100</i> <i>Total Credits:04</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) To get acquainted with the main components of an OS, and study concepts like system calls, processes management, threads, scheduling, synchronization, deadlocks, memory management, IO management.</li> <li>2) To understand the working of an OS as a resource manager, file system manager, process manager, memory manager and I/O manager and methods used to implement the different parts of OS.</li> <li>3) To study the need for special purpose operating systems with the advent of new emerging technologies</li> </ol>		
<b>Unit-I</b>	<b>[04]</b>	<b>Max Marks:08</b>
<b>Introduction:</b> review of computer organization, introduction to popular operating systems like UNIX, Windows, etc., OS structure, system calls, functions of OS, evolution of Oss.		
<b>Unit-II</b>	<b>[03]</b>	<b>Max Marks:06</b>
<b>Computer organization interface:</b> using interrupt handler to pass control between a running program and OS.		
<b>Unit-III</b>	<b>[08]</b>	<b>Max Marks:12</b>
<b>Concept of a process:</b> states, operations with examples from UNIX (fork, exec), Process scheduling, interprocess communication (shared memory and message passing), UNIX signals.		
<b>Unit-IV</b>	<b>[04]</b>	<b>Max Marks:06</b>
<b>Threads:</b> multithreaded model, scheduler activations, examples of threaded programs.		
<b>Unit-V</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Scheduling:</b> multi-programming and time sharing, scheduling algorithms, multiprocessor scheduling, thread scheduling (examples using POSIX threads) .		
<b>Unit-VI</b>	<b>[08]</b>	<b>Max Marks:12</b>
<b>Process synchronization:</b> critical sections, classical two process and n-process solutions, hardware primitives for synchronization, semaphores, monitors, classical problems in synchronization (producer-consumer, readers-writer, dining philosophers, etc.).		
<b>Unit-VII</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Deadlocks:</b> modelling, characterization, prevention and avoidance, detection and recovery.		
<b>Unit-VIII</b>	<b>[07]</b>	<b>Max Marks:12</b>
<b>Memory management:</b> with and without swapping, paging and segmentation, demand paging, virtual memory, page replacement algorithms, working set model, implementations from operating systems such as UNIX. Current Hardware support for paging: e.g., Pentium/ MIPS processor etc.		
<b>Unit-IX</b>	<b>[07]</b>	<b>Max Marks:12</b>
<b>Secondary storage and Input/Output:</b> device controllers and device drivers, disks, scheduling algorithms, file systems, directory structure, device controllers and device drivers, disks, disk space management, disk scheduling, NFS, RAID, other devices. Operations on them, UNIX FS, UFS protection and security, NFS		

<b>Unit-X</b>	<b>[04]</b>	<b>Max Marks:06</b>
<b>Protection and security:</b> Illustrations of security model of UNIX and other Oss. Examples of attacks.		
<b>Unit-XI</b>	<b>[03]</b>	<b>Max Marks:06</b>
<b>Epilogue:</b> Pointers to advanced topics (distributed OS, multimedia OS, embedded OS, real-time OS, OS for multiprocessor machines).		
<b>All above topics shall be illustrated using UNIX as case-studies.</b>		
<b>References:</b> <ol style="list-style-type: none"> <li>1) Abraham Silberschatz, Peter B. Galvin, Greg Gagne (2009), Operating System Concepts, 8<sup>th</sup> Ed., John Wiley ISBN 0-471-69466-5.</li> <li>2) William Stallings (2014), Operating Systems: Internals and Design Principles. Pearson, 8<sup>th</sup> Ed. ISBN-13: 978-0-13-230998-1</li> <li>3) AS Tanenbaum (2009), Modern Operating Systems, 3<sup>rd</sup> Ed., Pearson. ISBN: 0135013011</li> <li>4) AS Tanenbaum, AS Woodhull (2006), Operating Systems Design and Implementation, 3<sup>rd</sup> Ed., Prentice Hall ISBN-10: 0131429388</li> <li>5) M. J. Bach (1986), Design of the Unix Operating System, Prentice Hall of India ISBN0. -13-201757-1 025</li> </ol>		
<b>Course Outcome:</b> After completion of this course, students shall be able to: <ol style="list-style-type: none"> <li>1. Analyze design aspects and data structures/policies/algorithms used for file subsystem, memory subsystem, process subsystem and I/O subsystem of Unix OS.</li> <li>2. Differentiate between threads and processes and compare different processor scheduling algorithms.</li> <li>3. Identify the need to create the advance and special purpose operating system.</li> </ol>		

<b>Course Code: MCA-413</b>	<b>Fundamentals of Artificial Intelligence</b>	<b>Clock Hours: 60</b> <b>Total Marks: 100</b> <b>Total Credits:04</b>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) Gain a historical perspective of AI and its foundations.</li> <li>2) Study the concepts of Artificial Intelligence and investigate applications of AI techniques in intelligent agents.</li> <li>3) Learn various peculiar search strategies used in AI and use of them in solving problems using Artificial Intelligence.</li> </ol>		
<b>Unit-I</b>	<b>[08]</b>	<b>Max Marks:10</b>
<b>What is AI?:</b> Overview and Historical Perspective, Turing test, Physical Symbol Systems and the scope of Symbolic AI, AI Agents.		
<b>Unit-II</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Uninformed Search:</b> State Space Representation, Depth First Search, Breadth First Search, DFID.		
<b>Unit-III</b>	<b>[08]</b>	<b>Max Marks:12</b>
<b>Informed Search:</b> Best First Search, Hill Climbing, Beam Search, Tabu Search.		
<b>Unit-IV</b>	<b>[08]</b>	<b>Max Marks:15</b>
<b>Randomized Search:</b> Simulated Annealing, Genetic Algorithms, Ant Colony Optimization.		
<b>Unit-V</b>	<b>[08]</b>	<b>Max Marks:12</b>
<b>Problem Decomposition:</b> Goal Trees, AO*, Rule Based Systems, Rete Net.		
<b>Unit-VI</b>	<b>[06]</b>	<b>Max Marks:12</b>
<b>Game Playing:</b> Minimax Algorithm, AlphaBeta Algorithm, SSS*.		
<b>Unit-VII</b>	<b>[08]</b>	<b>Max Marks:14</b>
<b>Mathematical Logic and Inferences:</b> Propositional Logic, First Order Logic, Soundness and Completeness, Forward and Backward chaining.		
<b>Unit-VIII</b>	<b>[08]</b>	<b>Max Marks:15</b>
<b>Planning:</b> Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graph plan,		
<b>References:</b> <ol style="list-style-type: none"> <li>1) Deepak Khemani (2013). A First Course in Artificial Intelligence, McGraw Hill Education (India).</li> <li>2) Elaine Rich and Kevin Knight (1991). Artificial Intelligence, Tata McGraw Hill.</li> <li>3) Stuart Russell and Peter Norvig (2009). Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall.</li> </ol>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"> <li>1. Identify problems that are amenable to solution by AI methods.</li> <li>2. Identify appropriate AI methods to solve a given problem.</li> <li>3. Design smart system using different informed search / uninformed search or heuristic approaches.</li> </ol>		

<i>Course Code: MCA-414</i>	<b>Object Oriented Programming using C++</b>	<i>Clock Hours: 60 Total Marks: 100 Total Credits: 04</i>
<b>Course Objectives:</b> The objectives of the course are: <ol style="list-style-type: none"> <li>1) To familiarize the Object-Oriented Programming (OOP) concepts, such as abstraction, encapsulation, instances, initializations, polymorphism, overloading, inheritance etc.</li> <li>2) To write programs to solve problems using generic programming constructs such as templates and using standard template library.</li> <li>3) To understand and know the importance of pointers and learn file handling and exception handling in real-world problems.</li> </ol>		
<b>Unit-I</b>	<b>[15]</b>	<b>Max Marks: 15</b>
<b>Fundamentals:</b> <i>Object-Oriented Programming (OOP):</i> Need, Object Oriented Programming Paradigm, Benefits of OOP, C++ as object-oriented programming language. <i>C++ programming Basics:</i> Data types, Enumerations, Arrays, Strings, Pointers and references, Control structures. <i>Functions:</i> Function prototypes, parameter lists and return values, default values, global scoping, referencing, the 'const' keyword, referencing of strings, constant pointers, inline functions, static functions, function overloading, friend functions. <i>OOP Concepts:</i> The 'Struct' keyword, Functions within structures, Data encapsulation and classes, 'this' pointer, Constructors and Destructors, Overloading constructors, Copy Constructor, Assignment and Copy Initialization, Methods and their return values, Objects and Memory requirements, Static Class members, friend class.		
<b>Unit-II</b>	<b>[10]</b>	<b>Max Marks: 10</b>
<b>Inheritance:</b> Base Class and derived Class, access specifiers, Constructor and Destructor in Derived Class, Virtual destructor, Protected members, Overriding member functions, Public and Private Inheritance, Multiple Inheritance, Ambiguity in Multiple Inheritance, Composition, Nested Classes.		
<b>Unit-III</b>	<b>[10]</b>	<b>Max Marks: 20</b>
<b>Polymorphism:</b> <i>Operator Overloading:</i> concept of overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Keywords 'explicit' and 'mutable'. <i>Pointers-</i> indirect ion Operators, Memory Management: new and delete, Pointers to Objects. <i>Virtual Functions:</i> concept, pure virtual functions and abstract classes, arrays in polymorphism, late binding, Function pointers, Debugging Pointers, Dynamic Pointers, smart pointers.		



<b>Unit-IV</b>	<b>[10]</b>	<b>Max Marks:20</b>
<b>Files and Streams:</b> Data hierarchy, Stream Classes, Stream Errors, Disk File I/O with Streams, File Pointers, and Error Handling in File I/O, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Object, Command-Line Arguments, Printer output.		
<b>Unit-V</b>	<b>[10]</b>	<b>Max Marks:20</b>
<b>Templates and Exception Handling:</b> <i>Templates:</i> Function templates, Template specialization, Class templates, Non-type parameters for templates, template, and inheritance, The typename and export keywords. <i>Exception Handling:</i> Other error handling techniques, Exceptions, Exception handling in C++, rethrowing an exception, exception specifications, processing unexpected exceptions, stack unwinding, exception handling in constructors, destructors.		
<b>Unit-VI</b>	<b>[05]</b>	<b>Max Marks:15</b>
<b>Standard Template Library (STL):</b> <i>Introduction to STL:</i> Containers, algorithms, adaptors, and iterators, <i>Containers:</i> Sequence container and associative containers, <i>Adaptors:</i> container adapters, iterator adapters, <i>Algorithms:</i> basic searching and sorting algorithms, min-max algorithm, set operations, <i>Iterators:</i> input, output, forward, bidirectional and random access.		
<b>References:</b> 1) Robert Lafore, Object-Oriented Programming in C++, fourth edition, Sams Publishing, ISBN:0672323087. 2) Bjarne Stroustrup, The C++ Programming language, Third edition, Pearson Education ISBN 0-201-88954-4. 3) Meeta Gandhi, Tilak Shetty, Rajiv Shah, Vijay Mukhi's The 'C' Odyssey C++ and Graphics- The future of C, BPB publications, First Edition		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"> <li>1. Understand and use the basic programming constructs of C++ and manipulate various C++ datatypes, such as arrays, strings, and pointers.</li> <li>2. Manage memory appropriately using proper allocation/deallocation procedures.</li> <li>3. Write small-scale C++ programs using the above skills.</li> </ol>		

<b>Course Code: MCA-415</b>	<b>LAB on DBMS</b>	<b>Clock Hours: 60</b> <b>Total Marks: 50</b> <b>Total Credits:02</b>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) Provides foundation knowledge in database concepts, technology, and practice to prepare students into expert database application developers.</li> <li>2) Strong practice in SQL programming through a variety of database problems.</li> <li>3) Develop database applications using front-end tools and back-end DBMS.</li> </ol>		
<ol style="list-style-type: none"> <li>1) Creating database tables and using data types.</li> <li>2) Create table , Modify table, Drop table</li> <li>3) Practical Based on Data Manipulation. <ol style="list-style-type: none"> <li>a. Adding/Modify/Delete data using Insert/ Update/ Delete</li> </ol> </li> <li>4) Practical Based on Implementing the Constraints. <ol style="list-style-type: none"> <li>a. NULL and NOT NULL, Primary Key Constraint, Foreign Key Constraint</li> <li>b. Unique Constraint, Check Constraint, Default Constraint</li> </ol> </li> <li>5) Practical for Retrieving Data Using following clauses. <ol style="list-style-type: none"> <li>a. Simple select clause</li> <li>b. Accessing specific data with Where Clause</li> <li>c. Ordered By/ Distinct/Group By Clause</li> </ol> </li> <li>6) Practical Based on Aggregate Functions. <ol style="list-style-type: none"> <li>a. AVG, COUNT, MAX, MIN, SUM, CUBE</li> </ol> </li> <li>7) Practical Based on implementing all String functions.</li> <li>8) Practical Based on implementing Date and Time Functions.</li> <li>9) Practical Based on implementing use of UNION, INTERSECTION, SET DIFFERENCE.</li> <li>10) Implement Nested Queries &amp; all types of JOIN operation.</li> <li>11) Practical Based on performing different operations on a view.</li> <li>12) Practical Based on implementing use of Procedures.</li> <li>13) Practical Based on implementing use of Triggers.</li> <li>14) Practical Based on implementing Cursor.</li> <li>15) Demonstrate Database connectivity with front end tools like – VB.NET, C#.NET, JAVA etc.</li> <li>16) Practical based on creating Data Reports.</li> <li>17) Design entity relationship models for a business problem and develop a normalized database structure.</li> </ol>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"> <li>1. Design and implement a database schema for a given problem-domain.</li> <li>2. Create and maintain tables using PL/SQL, Populate and query a database using SQL DML/DDDL commands and programming PL/SQL including stored procedures, stored functions, cursors, triggers.</li> <li>3. Application development using PL/SQL &amp; front-end tools.</li> </ol>		

<i>Course Code: MCA-417</i>	<b>LAB on OS (Linux)</b>	<i>Clock Hours: 60</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
-----------------------------	--------------------------	---

**Course Objectives:**

- 1) To understand the Installation of Linux system.
- 2) To understand and make effective use of Linux utilities and shell scripting language to solve problems.
- 3) To know basics of system administration tasks, installation, configuration, and administration of internet servers.

1. Demonstration on Installation of Linux system Direct Installation; Partitioning the Hard drive for Linux, Using Live CD, Virtual Machine, init and run levels.
2. Linux Commands and Shell Programming
3. Creating Users Accounts and Groups, Starting and Stopping Services, Files and File System (File Types and Permissions, Links, Size and Space, Date and Time), Working with Files: Reading Files, Searching for files, Copying, Moving, Renaming, Deleting, Linking, and Editing Files, Other Commands: ls, rm, rmdir, pwd, more, less. grep, sort, cat, head, tail, wc, tee, ps, top, tar, unzip, nice, kill, netstat, Disk related commands, checking disk free spaces read command, conditional and looping statements, case statements, parameter passing and arguments, Shell variables, system shell variables, shell keywords, Shell programs for performing various tasks (List to be given by the course instructor)
4. System Administration  
Managing user accounts-adding & deleting users, changing permissions and ownerships, Creating and managing groups, modifying group attributes, Temporary disable user's accounts, creating and mounting file system, checking and monitoring system performance file security & Permissions, becoming super user using su; Getting system information Backup and restore files, reconfiguration hardware with kudzu, installing and removing packages in Linux. X-Windows administration
5. Installation, Configuration and Administration of Internet Servers
  - Simple LAN
  - Proxy server(Squid), DNS server(BIND)
  - Mail server
  - Web server(Apache)
  - File server(Samba)
  - DHCP server, SSH server and client FTP server and client

**Course Outcomes:**

After completion of this course students shall be able to-

1. Implement the Installation of Linux system.
2. Understand the basic commands of Linux operating system and can write shell scripts.
3. Implement system administration tasks, installation, configuration, and administration of internet servers.

<i>Course Code: MCA-418</i>	<b>LAB on C++ Programming</b>	<i>Clock Hours: 60</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) Course Objectives:</li> <li>2) Apply object-oriented approaches to software problems in C++.</li> <li>3) Apply exception handling techniques to software problems in C++.</li> <li>4) Apply generic programming approaches using templates and efficiently use standard template library in software development.</li> </ol>		
<ol style="list-style-type: none"> <li>1. Write program to demonstrate class, use of constructor, constructor overloading and destructor.</li> <li>2. Write program to demonstrate use of arrays, strings, pointers, constants, and references.</li> <li>3. Write program to demonstrate use of operator overloading.</li> <li>4. Write program(s) to demonstrate the use of inheritance.</li> <li>5. Write program to demonstrate use of compile time and runtime polymorphism.</li> <li>6. Write program to demonstrate use of friend function and friend class.</li> <li>7. Write program to demonstrate use of virtual class.</li> <li>8. Write a program to demonstrate the use of static data member and static member function.</li> <li>9. Write a program to demonstrate file handling.</li> <li>10. Write program to demonstrate use of function templates.</li> <li>11. Write program to demonstrate use of class templates.</li> <li>12. Write program to demonstrate use of exception handling.</li> <li>13. Write a program to demonstrate command line arguments.</li> <li>14. Write program(s) to demonstrate use of STL.</li> </ol>		
<b>Course Outcomes:</b> <p>After completion of this course students shall be able to-</p> <ol style="list-style-type: none"> <li>1. Understands the fundamentals of C++ programming language and its constructs.</li> <li>2. Implement the applications using object-oriented programming concepts.</li> <li>3. Understand how to implement file handling exception handling and standard template library for C++ application development.</li> </ol>		

<i>Course Code: MCA-419 (A)</i>	<b>Computer Programming and Problem Solving (CPPS)</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) To introduce the foundations of computing, programming and problem-solving.</li> <li>2) To develop logical ability for problem-solving.</li> <li>3) To develop basic programming skills necessary for coding.</li> </ol>		
<b>Unit-I</b>	<b>[04]</b>	<b>Max Marks:06</b>
<b>Introduction to problem solving:</b> Problem solving aspect, Designs(top down and bottom up, functional programming, data storage and manipulations, classic puzzles, general problem solving techniques, expressing using charts, algorithms, introduction to pseudocode.		
<b>Unit-II</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Solving Problems with iterations verses Recursion:</b> Iterations- Review, problem solving with iterations, Review of Recursion Fundamentals, Head and Tail Recursion, Applying to Dynamic Data Storage, Recursion and Binary Trees, Wrapper Functions, When to Choose Recursion, Converting recursion to iterative.		
<b>Unit-III</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Solving Problems with Vector and Matrices:</b> Review of Array Fundamentals, Store, Copy, Retrieval and Search, Sort, Compute Statistics, Solving Problems with Arrays, Finding the Mode, Refactoring, Arrays of Fixed Data, Non-scalar Arrays , Multidimensional Arrays, Deciding When to Use Arrays.		
<b>Unit-IV</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Solving Problems with Dynamic Memory:</b> Benefits of using dynamic memory , Runtime-Sized Data Structures, Resizable Data Structures, Memory Sharing, When to Use dynamic memory, The Stack and the heap memory, Memory Size and its Lifetime, Variable-Length Strings, Solving Pointer Problems, Linked representations		
<b>Unit-V</b>	<b>[04]</b>	<b>Max Marks:08</b>
<b>Structural approach verses object oriented approach:</b> Introduction to object oriented paradigms, Components of Structural approach and object oriented approach, Structural approach verses object oriented approach		
<b>Unit-VI</b>	<b>[04]</b>	<b>Max Marks:06</b>
<b>Solving Problems with code reuse:</b> Good Reuse and Bad Reuse, Review of Component Fundamentals, Code Block and algorithms, Abstract Data Types, Patterns, Libraries, Building Component Knowledge		

**References:**

- 1) R. J. Dromey, "How to solve it by Computer" Prentice-Hall ISBN 978-0134340012
- 2) V. Anton Spraul "Think Like a Programmer: An Introduction to Creative Problem Solving", No Starch Press, Inc. ISBN: 978-1593274245
- 3) Subhashis Banerjee, S. Arun-Kumar, D. Dubhashi: Introduction to Computer Science. Manuscript.
- 4) Structure and Interpretation of Computer Programs by Harold Abelson and Gerald Sussman with Julie Sussman, MIT Press, 1985.

**Course Outcome:**

After completion of this course students shall be able to-

1. Design blocks of the problems.
2. Build logic for solving new problems on paper.
3. Model the logic as code.

<i>Course Code:</i> <b>MCA-419(B)</b>	<b>LAB on Computer Programming and Problem Solving (CPPS)</b>	<i>Clock Hours:</i> <b>60</b> <i>Total Marks:</i> <b>50</b> <i>Total Credits:</i> <b>02</b>
<b>Course Objectives:</b> <ol style="list-style-type: none"><li>1) To represent the problems using mechanisms like charts.</li><li>2) To express logic as an algorithm.</li><li>3) To convert logic in programs.</li></ol>		
<b>Laboratory Requirements:</b> <b>OS:</b> Windows/Linux, <b>Programming Language:</b> Any programming language <b>Instructions</b> <ul style="list-style-type: none"><li>• All practical assignments must be designed on paper, logic should be demonstrated, and pseudocode is to be written.</li><li>• No barrier of programming language for code conversion of the assignments.</li><li>• Problems and puzzles in theory are the assignments for the followings.</li></ul>		
<ol style="list-style-type: none"><li>1. Simple exercises and examples to introduce to the computing environment and usage.</li><li>2. Simple exercises and examples of functional programming</li><li>3. Puzzle solving using iterations</li><li>4. Problem solving using recursion</li><li>5. Programming for vectors and multidimensional data</li><li>6. Dynamic memory and problem solving.</li><li>7. Assignments on Solving Problems with code reuse</li></ol>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"><li>1. Construct logic for the problems.</li><li>2. Write algorithms and be able to draw logic on paper.</li><li>3. Write code for the logic developed.</li></ol>		

<i>Course Code: MCA-420(A)</i>	<b>Web Programming</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) To understand web designing using HTML/CSS.</li> <li>2) To use JavaScript for scripting.</li> <li>3) To understand static/dynamic API using JSON/JQuery/AngularJS.</li> </ol>		
<b>Unit-I</b>	<b>[05]</b>	<b>Max Marks:10</b>
<b>HTML and CSS:</b> Introduction to HTML Introduction to CSS, Introduction to HTML5 <b>JavaScript:</b> Introduction to JavaScript, Variable, statements, Operators, Comments, constructs, Functions, expressions, JavaScript console, Scope, Events, Strings, String Methods, Numbers, Number Methods, Dates, Date Formats, Date Methods, Arrays, Loops Object Prototypes, Object Oriented Programming, JavaScript Validations, Security in Java Script		
<b>Unit-II</b>	<b>[05]</b>	<b>Max Marks:10</b>
<b>Bootstrap :</b> Introduction to Responsive Web Design, Overview of Bootstrap, Need to use Bootstrap, Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid, Typography, Tables, Images, Jumbotron, Wells, Alerts, Buttons, Button Groups, Badges/Labels, Progress Bars, Pagination, List Groups, Panels, Dropdowns, Collapse, Tabs/Pills, Navbar, Forms, Inputs, Bootstrap Grids, Grid System, Stacked/Horizontal, Bootstrap Themes, Templates		
<b>Unit-III</b>	<b>[05]</b>	<b>Max Marks:10</b>
<b>AngularJS:</b> Introduction to AngularJS, Structuring AngularJS application, MVC in AngularJS, AngularJS routing, AngularJS services		
<b>Unit-IV</b>	<b>[05]</b>	<b>Max Marks:10</b>
<b>JQuery:</b> Basics of jQuery, jquery selection and events, jQuery Effects, jquery traversal and manipulation, Data attributes and templates, jQuery Plugins, JQuery / Google Web Toolkit		
<b>Unit-V</b>	<b>[05]</b>	<b>Max Marks:05</b>
<b>Node.js:</b> Node.js: Introduction to Node.js, Node modules, Developing node.js web application, Event-driven I/O server-side JavaScript, Express: Introduction to Express, First Express Application, Application, Request and Response Objects, Implementing MVC Pattern, Express application configuration, Rendering Views.		
<b>Unit-VI</b>	<b>[05]</b>	<b>Max Marks:05</b>
<b>JSON:</b> Introduction, Need of JSON, JSON Syntax Rules, JSON Data - a Name and a Value, JSON Objects, JSON Arrays, JSON Uses JavaScript Syntax, JSON Files, JSON & Security Concerns, Cross Site Request Forgery (CSRF), Injection Attacks ,Responsive Web Design		
<b>References:</b> <ol style="list-style-type: none"> <li>1) HTML5 Programmer's Reference. Reid, J. (2015). Apress.</li> <li>2) Bootstrap: Responsive Web Development. Spurlock, J. (2013). O'Reilly Media.</li> <li>3) Professional AngularJS. Karpov, V., Netto, D. (2015). Wiley.</li> <li>4) Web Development with JQuery. York, R. (2015). Wiley.</li> <li>5) Professional Node.js: Building Javascript Based Scalable Software. Teixeira, P. (2012). Wiley.</li> <li>6) Beginning JSON. Smith, B. (2015). Apress.</li> </ol>		

**Course Outcome:**

After completion of this course students shall be able to-

1. Design the web applications/sites.
2. Apply dynamic paging using AngularJS/JSON/JQuery..
3. Use Javascript/Node.JS to make design and scripting.

<b>Course Code: MCA-420 (B)</b>	<b>LAB on Web Designing</b>	<b>Clock Hours: 60</b> <b>Total Marks: 50</b> <b>Total Credits: 02</b>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) To work in web designing using HTML/CSS.</li> <li>2) To use Bootstrap for designing.</li> <li>3) To design dynamically using JSON/JQuery/AngularJS.</li> </ol>		
<ol style="list-style-type: none"> <li>1. Design a website with HTML Form.</li> <li>2. Design a website using CSS 2.1 and CSS3.</li> <li>3. Design a website with HTML5.</li> <li>4. Design a dynamic web form with validations using JavaScript.</li> <li>5. Design a website with Bootstrap.</li> <li>6. Design a dynamic website with AngularJS.</li> <li>7. Demonstrate the use of jQuery in a website.</li> <li>8. Demonstrate the use of Node.js in a website.</li> <li>9. Demonstrate the use of JSON in a website.</li> <li>10. Design a dynamic website using demonstrating the web technologies (HTML, JavaScript, Bootstrap, Angular JS, JQuery).</li> </ol>		
<b>Course Outcome:</b> <p>After completion of this course students shall be able to-</p> <ol style="list-style-type: none"> <li>1. Develop Web site/App.</li> <li>2. Use Bootstrap/Javascript to make design and scripting.</li> <li>3. Make Web site dynamic using AngularJS/JSON/JQuery.</li> </ol>		



## Semester-II

<i>Course Code: MCA-421</i>	<b>Advanced Software Development Methodologies</b>	<i>Clock Hours: 60</i> <i>Total Marks: 100</i> <i>Total Credits:04</i>
<b>Course Objectives:</b> The objectives of the course are: <ol style="list-style-type: none"> <li>1) To introduce git for software development</li> <li>2) To learn the principles and practices associated with each of the agile development methods.</li> <li>3) To apply the principles and practices of agile software development on a project of interest and relevance to the student.</li> </ol>		
<b>Unit-I</b>	<b>[10]</b>	<b>Max Marks:16</b>
<b>Git &amp; Version Control:</b> Basic Concepts, Environment setup, Life Cycle, Branches & Merging, working with local repository and Remote Repository.		
<b>Unit-II</b>	<b>[10]</b>	<b>Max Marks:18</b>
<b>Agile Methodology:</b> Introduction, software development with agile, traditional model vs agile model, agile methods classification, manifesto and principles, project management, team interactions, ethics in teams, agility in design and testing, documentations, agile drivers, capabilities and values.		
<b>Unit-III</b>	<b>[12]</b>	<b>Max Marks:20</b>
<b>Agile Processes:</b> Lean production - SCRUM, Crystal, Feature Driven Development, Adaptive Software Development, and Extreme Programming: Method overview, lifecycle, work products, roles and practices.		
<b>Unit-IV</b>	<b>[12]</b>	<b>Max Marks:18</b>
<b>Agility and Knowledge Management:</b> Agile information systems, decision making, knowledge management, institutional knowledge evolution cycle, development, acquisition, refinement, distribution, deployment, leveraging, knowledge management in software engineering, managing software knowledge, challenges of migrating to agile methodologies, agile knowledge sharing, story-cards and Story-card Maturity Model (SMM).		
<b>Unit-V</b>	<b>[08]</b>	<b>Max Marks:14</b>
<b>Agility and Requirements Engineering:</b> Impact of agile processes, current practices, variance, requirement engineering, managing unstable requirements, requirements elicitation, abstraction model, requirements management in agile environment, requirements prioritization, requirements modeling and generation, concurrency.		
<b>Unit-VI</b>	<b>[08]</b>	<b>Max Marks:14</b>
<b>Agility and Quality Assurance:</b> Agile Interaction Design and product development, Agile Metrics and Feature Driven Development (FDD), Financial and Production Metrics in FDD, Quality Assurance with Agile approach, Test Driven Development, Pair programming, Global Software Development.		

**References:**

- 1) Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices Alan Apt Series.
- 2) Succeeding with Agile: Software Development Using Scrum, Pearson.
- 3) [www.github.com](http://www.github.com)

**Course Outcome:**

After completion of this course students shall be able to-

1. Use git for software development and deployment.
2. Apply a thorough understanding of Agile principles and specific practices.
3. Judge, craft and evaluate appropriate adaptations to existing practices or processes depending upon analysis of typical problems.

<i>Course Code: MCA-422</i>	<b>Data Structures and Algorithms</b>	<i>Clock Hours: 60</i> <i>Total Marks: 100</i> <i>Total Credits:04</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) To impart the basic concepts of data structures and algorithms</li> <li>2) To understand basic concepts about array, stacks, queues, linked lists, trees and graphs and advance topics like AVL Trees, BTrees, B* and B+ Trees</li> <li>3) To understand concepts about searching, sorting and hashing techniques</li> </ol>		
<b>Unit-I</b>	<b>[08]</b>	<b>Max Marks:06</b>
<b>Introduction to Data Structures and Algorithms:</b> Algorithmic Notation: Format Conventions, Statement and Control Structures. Time and Space Analysis: Data types and Abstract data types, Types of Data structures; Primitive, Non primitive, Linear and Nonlinear Data structures		
<b>Unit-II</b>	<b>[08]</b>	<b>Max Marks:15</b>
<b>Array:</b> Storage representation, operations and applications (Polynomial addition and subtraction) <b>Stack:</b> operations and applications (infix, postfix and prefix expression handling), <b>Queue:</b> operations and applications, Circular Queues: operations and applications, Concept of Double ended Queue and Priority Queue, Linked representation of stack and queue.		
<b>Unit-III</b>	<b>[10]</b>	<b>Max Marks:12</b>
<b>Linked Lists:</b> Operations and Applications of Linear linked list (Polynomial addition and subtraction), Circular linked list and Doubly linked list.		
<b>Unit-IV</b>	<b>[11]</b>	<b>Max Marks:21</b>
<b>Trees:</b> Binary Trees, Binary Tree: Representations, Operations (insert/delete), Traversal (inorder, preorder, postorder, level order), Threaded Binary Tree, Search Trees: AVL Tree, single and double rotations, M-Way Search Tree (definition), B-Trees: insertion and deletion operation		
<b>Unit-V</b>	<b>[11]</b>	<b>Max Marks:18</b>
<b>Graphs and Their Applications:</b> Representation (Matrix/Adjacency) and Traversal (Depth First Search/Breadth First Search), Spanning Trees, Minimal Spanning Tree (Prim's and Kruskals's algorithm), Shortest Paths and All Pair Shortest Path: Dijkstra's, Floyd-Warshall Algorithms.		
<b>Unit-VI</b>	<b>[12]</b>	<b>Max Marks:18</b>
<b>Hash Table:</b> Hash Function, Collision and its Resolution, Separate Chaining, Open Addressing (linear probing, quadratic probing, double hashing), Rehashing, Extendible Hashing <b>Searching:</b> Linear Search and Binary Search (array/binary tree). <b>Sorting:</b> General Background, Sorting Techniques: Bubble Sort, Insertion Sort, Selection Sort, Quicksort, Mergesort, Heapsort and Radix Sort.		

**References:**

- 1) Tremblay, J. & Sorenson, P.G., (2001), An Introduction to Data Structures with Application, Mcgraw Hill India, ISBN: 978-0074624715, 0074624717
- 2) Langsam, Y., Augenstein, M.J. & Tenenbaum A.M., (2015), Data Structures using C and C++, 2<sup>nd</sup> Edition, Pearson Education ISBN: 978-9332549319, 9332549311
- 3) Balagurusamy, E., (2013), Data Structures using C, 1<sup>st</sup> Edition, Mcgraw Hill Education, ISBN: 978-1259029547, 1259029549
- 4) Weiss, M.A., (2002), Data Structures and Algorithm Analysis in C, 2<sup>nd</sup> Edition, Pearson India, ISBN: 978-8177583588, 8177583581
- 5) Horowitz, E., Sartaj S. & Mehta, D. (2008), Fundamentals of Data Structures in C++, Universities Press ISBN: 978-8173716065, 8173716064
- 6) Lafore, R., (2003), Data Structures & Algorithms in Java, 2<sup>nd</sup> Edition, Pearson India, ISBN: 978-8131718124, 8131718123
- 7) Kruse, R., Tondo, C.L., Leung B., & Mogalla S, (2006), Data Structures and Program Design in C, Pearson India, ISBN: 978-8177584233.

**Course Outcome:**

After completion of this course students shall be able to-

1. Understand the concept of Dynamic memory management, data types, algorithms, Big O notation.
2. Understand data structures such as arrays, linked lists, stacks and queues, graphs, trees and hash tables.
3. Solve problem involving graphs, trees and apply different sorting and searching algorithms.

<i>Course Code: MCA-423</i>	<b>Java Programming (Core Java)</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) Understand Fundamental concepts of object-oriented programming using Java technology.</li> <li>2) Java applications development using polymorphism, inheritance, interfaces and inner classes and multi-threading.</li> <li>3) GUI applications and event driven applications development.</li> </ol>		
<b>Unit-I</b>	<b>[04]</b>	<b>Max Marks:5</b>
<b>An Introduction to Java:</b> History of Java, Features of Java ( Java Buzz words), Obtaining Java Environment, Setting up Java Environment, Structure of the Java Program, Creating a Source File, Compiling the Source File into a .class file, Executing the Program, The Java Virtual Machine, Comments, Data types, variables, Keywords, Operators, Control Structures, Arrays		
<b>Unit-II</b>	<b>[04]</b>	<b>Max Marks:5</b>
<b>Introduction to OOPs:</b> OOPs concepts, Predefined classes(String, StringBuffer), type casting, wrapper classes, Input and Output, User defined class, object creation and initialization, finalize() method, static fields and methods, this keyword, Access specifier Inner class		
<b>Unit-III</b>	<b>[05]</b>	<b>Max Marks:10</b>
<b>Inheritance, Polymorphism and interfaces:</b> Dynamic Polymorphism (Method Overloading and Method Overriding), Static Polymorphism, final keyword, Superclass, Subclass, super keyword, Abstract classes, Methods with a Variable Number of Parameters, Enumeration Classes, Interfaces, Reflection		
<b>Unit-IV</b>	<b>[05]</b>	<b>Max Marks:10</b>
<b>Multithreading and Exceptions:</b> Creating Thread, Multi-Tasking using Threads, Thread Synchronization or Thread Safe, Thread Class Methods, Thread Communication, Thread Properties, ThreadGroup, Thread States (Life-Cycle of a Thread), Exception handling ( try, catch, finally), throws clause, throw clause, Types of Exceptions(built-in, user defined), Assertions		
<b>Unit-V</b>	<b>[08]</b>	<b>Max Marks:12</b>
<b>Graphics Programming and event handling :</b> Introduction to swing and awt, Creating a Frame, Positioning a Frame, Displaying Information in a Component, Working with 2D Shapes, Color, Special Fonts for Text, JComponent class Methods, Creating Components in Swing ( PushButton, Label, JComboBox Class , JList Class, JMenu Class), Layout Manager, Basics of Event Handling, Listeners and Listener Methods, Mouse Events, Keyboard Events, AWT Event Hierarchy		
<b>Unit-VI</b>	<b>[04]</b>	<b>Max Marks:08</b>
<b>Streams, Files and JDBC:</b> Input and output stream, Reading and Writing Binary Data, Reading and Writing text Data, File Management (File Class), The Design of JDBC, JDBC Configuration, Executing SQL Statements, Query Execution Scrollable and Updatable Result Sets, Row Sets, Metadata, Transactions, Packages.		

***References:***

- 1) Horstman Cay, Cornell Gary, Core JavaTM2, Vol.1&2, Seventh Edition, Pearson education.
- 2) Herbert Schildt, The Complete Reference, Seventh Edition, Tata McGraw-Hill.
- 3) Steven Holzner, JAVA 2 Programming Black Book, Wiley India. 4] Ivor Horton, Beginning Java 2, JDK 5 Ed, Wiley India.

***Course Outcome:***

After completion of this course students shall be able to -

1. Create Java application development using polymorphism, inheritance, and inner classes.
2. Develop GUI interface and event driven applications.
3. Manipulate databases through java application.

<i>Course Code: MCA-424</i>	<b>LAB on Data Structures and Algorithms</b>	<i>Clock Hours: 60</i> <i>Total Marks: 50</i> <i>Total Credits: 02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) Solve real-world problems by reasoning about data structure choices, choose appropriate implementations.</li> <li>2) To make the students write various programs and ADTS for all data structures.</li> <li>3) Students will learn to write, debug, and test large programs systematically.</li> </ol>		
<b>Implementation of programs based on the following</b> <ul style="list-style-type: none"> <li>• Arrays</li> <li>• Multidimensional Arrays, Matrices</li> <li>• Stacks, Polish Notation</li> <li>• Queues</li> <li>• Deques</li> <li>• Linear Linked List, Circular Linked List, Doubly Linked List</li> <li>• Polynomial Addition/Subtraction</li> </ul>		
<b>Implementation of programs based on Trees</b> <ul style="list-style-type: none"> <li>• Binary Search Tree</li> <li>• In-order, Pre-order and Post-order Traversals</li> <li>• Heap Tree</li> </ul>		
<b>Implementation of programs based on Graphs</b> <ul style="list-style-type: none"> <li>• Depth First Traversal</li> <li>• Breadth First Traversal</li> <li>• Obtaining Shortest Path (Dijkstra and Floyd-Warshall)</li> <li>• Minimum spanning tree (Kruskal and Prim)</li> </ul>		
<b>Implementation of programs for Hash Table, Searching and Sorting techniques</b> <ul style="list-style-type: none"> <li>• Hash Table</li> <li>• Linear and Binary Search (using array)</li> <li>• Bubble sort</li> <li>• Selection sort</li> <li>• Insertion sort</li> <li>• Radix sort</li> <li>• Quick sort</li> <li>• Merge sort</li> <li>• Heap sort</li> </ul>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"> <li>1. Develop solutions for a range of problems using procedure oriented / object-oriented programming.</li> <li>2. Choose the appropriate data structure and algorithm design method for a specified application.</li> <li>3. Apply practical knowledge on the applications of data structures.</li> </ol>		

<i>Course Code: MCA-425</i>	<b>LAB on Java Programming</b>	<i>Clock Hours: 60</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) Solve real-world problems by reasoning about Java programming choices, choose appropriate implementations.</li> <li>2) To make the students write various programs in java technology.</li> <li>3) Students will learn to write, debug, and test large programs systematically.</li> </ol>		
<b>Implementation of programs based on the following</b> <ul style="list-style-type: none"> <li>• Write a program that demonstrates program structure of java with use of arithmetical and logical implementation.</li> <li>• Write a program that demonstrates string operations using String and StringBuffer class.</li> <li>• Write a program to demonstrate inner class and static fields.</li> <li>• Write a program that demonstrates inheritance, polymorphism.</li> <li>• Write a program that demonstrates 2D shapes on frames.</li> <li>• Write a program that demonstrates color and fonts.</li> <li>• Write a program to illustrate the use of various swing components.</li> <li>• Write a program that demonstrates use of dialog box and menus.</li> <li>• Write a program that demonstrates event handling for various types of events.</li> <li>• Write a program to illustrate multithreading.</li> <li>• Write a program to illustrate exception handling.</li> <li>• Write a program to demonstrate the use of File class.</li> <li>• Write a program that demonstrates JDBC on application.</li> <li>• Write a program that demonstrate package creation and use in program.</li> </ul>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"> <li>1. Develop solutions for a range of problems using object-oriented programming.</li> <li>2. Choose the appropriate data structure and algorithm design method for a specified application.</li> <li>3. Apply practical knowledge on the applications using Java.</li> </ol>		



<i>Course Code: MCA-426 (A)</i>	<b>Digital Image Processing (DIP)</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) The fundamental knowledge and basic technical competence in the field of Computer Graphics and Digital Image Processing.</li> <li>2) Give an in-depth knowledge about 2D and 3D transformation algorithms.</li> <li>3) Provide awareness about the current technologies and issues specific to Digital Image Enhancement, Restoration, Segmentation, Color Image Processing, and Morphological Image Processing.</li> </ol>		
<b>Unit-I Introduction to Digital Image Processing &amp; Applications</b>	<b>[06]</b>	<b>Max Marks:12</b>
Digital Image Processing. Applications of Digital Image Processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Image Sensing and Acquisition. Image Sampling and Quantization. Some Basic Relationships Between Pixels.		
<b>Unit-II Image Enhancement</b>	<b>[10]</b>	<b>Max Marks:13</b>
Background, Some Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Combining Spatial Enhancement Methods, Introduction to the Fourier Transform and the Frequency Domain, Smoothing Frequency-Domain Filters, Sharpening Frequency Domain Filters, Homomorphic Filtering,		
<b>Unit-III Color Image Processing</b>	<b>[04]</b>	<b>Max Marks:10</b>
Color Fundamentals, Color Models, Pseudocolor Image Processing, Basics of Full-Color Image Processing, Color Transformations, Smoothing and Sharpening.		
<b>Unit-IV Introduction to computer vision</b>	<b>[10]</b>	<b>Max Marks: 15</b>
Role of Artificial intelligence and image processing in Computer Vision, Industrial Machine Vision applications, System architecture. Visual Sensors: Camera sensors: RGB, IR, Kinect sensor, Camera interfaces and video standards, Characteristics of camera sensors commercially available cameras. Camera Calibration: Interior, exterior calibration and rectification using Tsai's Calibration method.		
<b>References:</b> <ol style="list-style-type: none"> <li>1) R.C.Gonzalez &amp; R.E.Woods, Digital Image Processing, Pearson Education, 3rd edition, ISBN. 13:978-0131687288</li> <li>2) S. Jayaraman Digital Image Processing TMH (McGraw Hill) publication, ISBN-13:978-0-07-0144798</li> <li>3) Gonzalez, Woods &amp; Steven, Digital Image Processing using MATLAB, Pearson Education, ISBN-13:978-0130085191</li> <li>4) .Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.</li> <li>5) Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.</li> </ol>		

**Course Outcome:**

After completion of this course students shall be able to-

1. Develop scientific and strategic approach to solve complex problems in the domain of Computer Graphics and Digital Image Processing; expose students to Open Source Image Processing software (OpenCV or Scilab etc)
2. Demonstrate various algorithms for scan conversion and filling of basic primitives objects and their comparative analysis and applied 2-D and 3-D geometric transformations, viewing and clipping on graphical objects.
3. Use the Mathematics for digital image representation, image acquisition, image transformation, image enhancement and restoration.

<b>Course Code: MCA-426 (B)</b>	<b>LAB on DIP</b>	<b>Clock Hours: 60</b> <b>Total Marks: 50</b> <b>Total Credits:02</b>
---------------------------------	-------------------	---

**Course Objectives:**

- 1) The fundamental knowledge and basic technical competence in the field of Computer Graphics and Digital Image Processing.
- 2) Give an in-depth knowledge about 2D and 3D transformation algorithms.
- 3) Provide awareness about the current technologies and issues specific to Digital Image Enhancement, Restoration, Segmentation, Color Image Processing, and Morphological Image Processing.

1. Read an 8 bit image and then apply different image enhancement techniques:
  - (a) Brightness improvement
  - (b) Brightness reduction
  - (c) Thresholding
  - (d) Negative of an image
  - (e) Log transformation
  - (f) Power Law transformation.
2. Read an image, plot its histogram then do histogram equalization. Comment about the result.
3. (a) Implement Gray level slicing (intensity level slicing) in to read cameraman image.  
(b) Read an 8 bit image and to see the effect of each bit on the image.  
(c) Read an image and to extract 8 different planes i.e. ‘bit plane slicing.’
4. Implement various Smoothing spatial filter.
5. Read an image and apply
  - (1) Gaussian 3x3 mask for blurring
  - (2) High pass filter mask with different masks
  - (3) Laplacian operator with centre value positive and negative
  - (4) High boost filtering.
6. Write a program to implement various low pass filters and high pass filter in frequency domain.

7. Implement and study the effect of Different Mask (Sobel, Prewitt and Roberts)
8. Write a program to implement Object localization and processing: Contours, edges, lines, skeletons.
9. Write a program to implement feature extraction, Feature Selection, Scene and Object Discrimination.
10. Write a program to human face tracking from given image.

***Course Outcome:***

After completion of this course students shall be able to-

1. Develop scientific and strategic approach to solve complex problems in the domain of Computer Graphics and Digital Image Processing;
2. Implement various algorithms for scan conversion, filling objects, 2-D and 3-D geometric transformations, viewing and clipping on graphical objects;
3. Make use of Open Source Image Processing software (Like OpenCV or Scilab) to implement image transformation, image enhancement in spatial and frequency domain.

<i>Course Code: MCA-427 (A)</i>	<b>Python Programming</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) Introduce to python programming: data types, operators, conditional and logical statements, control structures, writing user defined functions and file handling.</li> <li>2) To introduce to OOP through python, regular expressions, exception handling and GUI constructs for web.</li> <li>3) To study advance topics in python viz., lambda functions, functional programming tools, using and configuring modules etc.</li> </ol>		
<b>Unit-I</b>	<b>[06]</b>	<b>Max Marks:10</b>
The Python Programming, Features, Application, Variables, Identifier, Identifier Naming, Data Types, Comments in Python, Keywords, Literals, Type conversion, Functions, operators and its types, Order of Operations, Expressions, Scope of Variables, Functions, Defining Functions, Calling Functions, passing arguments in function, call by reference in python, Types of Arguments: required arguments, Formal Arguments, Default Arguments, Variable-length Arguments, Keyword Arguments, Built-in Functions, Decision Making: if statement, If..else statement, Chained conditionals, Loops: For loop, While loop, Loop control statements: break, continue, pass, Nested loop, Using else with for loop, Using else with while loop		
<b>Unit-II</b>	<b>[05]</b>	<b>Max Marks:10</b>
Strings: Creating string, indexing and splitting, accessing values in strings, reassigning strings, deleting string, Working with the Characters of a String, string operators, string formatting, Built-in String Methods, Length, The Slice Operator, String Comparison, Lists: Accessing Elements in list, list length, List Slices, list methods, list slices, List Membership, Concatenation and Repetition, Objects and References, Aliasing and Copying, Cloning Lists, list loop, mutability, List Deletion, Objects and References, aliasing and, cloning list, list as parameters, List Membership, Concatenation and Repetition, Append versus Concatenate Lists, Tuples: creating Tuple, Tuple indexing and slicing, Deleting Tuple, Tuple operations and built-in functions, List Vs Tuple, Tuples and Mutability, Tuple Assignment, Tuples as arguments, Tuples as Return Values.		
<b>Unit-III</b>	<b>[08]</b>	<b>Max Marks:10</b>
Dictionaries, Accessing the dictionary values, Adding dictionary values, Operations on Dictionary, Dictionary Methods, Built-in Dictionary methods, Iterating Dictionary, Dictionary Keys, Aliasing and Copying, Opening a file, The close() method, The with statement, Writing the file, Read file through for loop, Read Lines of the file, Creating a new file, File Pointer positions, Modifying file pointer position, Removing the file, Creating the new directory, Changing the current working directory, Deleting directory, The file related methods, Python Class and Objects, Creating classes in Python, Creating an instance of the class, Python Constructor, Types of Constructors, Python built-in class functions, Inheritance, Overloading Methods, Overriding methods, Data Hiding, Search Algorithms, Sorting Algorithms, Hash Tables		
<b>Unit-IV</b>	<b>[05]</b>	<b>Max Marks:10</b>

Regular Expressions, Exceptions, Standard Exceptions, Exceptions Syntax, The try/except/else Statement, The try/finally Statement, Unified try/except/finally, The raise Statement, The assert Statement, with/as Context Managers String-Based Exceptions, Class-Based Exceptions, General raise Statement Forms, Nesting Exception Handlers, Exception Idioms, Exception Design Tips. Catch All Exceptions, Catch A Specific Exception, Catch Multiple Specific Exceptions, Clean-up After Exceptions, GUI Programming using TKinter.

<b>Unit-V</b>	<b>[06]</b>	<b>Max Marks:10</b>
---------------	-------------	---------------------

Advance Function Topics: Anonymous Function Lambda, Mapping Functions over Sequences: map, Functional Programming Tools: filter and reduce, List Comprehensions Revisited: Mappings. Modules: Python Program Architecture, Module Creation, Module usage, Module Namespaces, Reloading Modules, Module Packages. Data Hiding in Modules, Enabling Future Language Features, Mixed Usage Modes, Changing the Module Search Path, The import as Extension, Relative Import Syntax, Module Design Concepts

**References:**

- 1) John V Guttag (2013), Introduction to Computation and Programming Using Python, Prentice Hall of India, 2013, ISBN: 9780262525008
- 2) R. Nageswara Rao(2016), Core Python Programming, Dreamtech Press, 2016, ISBN-13: 9789351199427
- 3) Wesley J. Chun(2006), Core Python Programming – Second Edition, Prentice Hall, ISBN- 13: 978-0132269933, ISBN-10: 0132269937
- 4) Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser(2013), Data Structures and Algorithms in Python”, Wiley, 2013, ISBN : 978-1-118-54958-2, ISBN : 978-1-118- 29027-9(HardCover)
- 5) Kenneth A. Lambert(2011), Fundamentals of Python – First Programs, CENGAGE Publication, 2011, ISBN 1111822700, ISBN 9781111822705
- 6) Luke Sneeringer(2015), Professional Python, Wiley Inc.,2015, ISBN: 1119070856
- 7) Mark Lutz (2007), Learning Python, 3<sup>rd</sup> Edition, O’Reilly Media, Inc., 2007, ISBN-13: 978-0-596-51398- 6, ISBN-10: 0-596-51398-4

**Course Outcome:**

After completion of this course students shall be able to-

1. Use lists, tuples, dictionaries, strings and files efficiently for solving real world problems.
2. Implement the concepts of object-oriented programming using python.
3. Develop modules, packages and GUI based programming for web.

<i>Course Code: MCA-427 (B)</i>	<b>LAB on Python programming</b>	<i>Clock Hours: 60</i> <i>Total Marks: 50</i> <i>Total Credits: 02</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1) To acquire programming skills in core Python.</li> <li>2) To develop the skill of designing Graphical user Interfaces in Python</li> <li>3) To develop the ability to write file handling, exception handling and modular programming applications in Python.</li> </ol>		
<ol style="list-style-type: none"> <li>1. Develop programs to understand the control structures of python</li> <li>2. Develop programs to learn different types of structures (list, dictionary, tuples) in python</li> <li>3. Develop programs to learn concept of functions scoping, recursion and list mutability.</li> <li>4. Develop programs to understand object oriented programming using python.</li> <li>5. Develop programs for data structure algorithms using python – searching, sorting and hash tables.</li> <li>6. Develop programs to learn regular expressions using python.</li> <li>7. Demonstrate the concept of exception handling using try/except/else Statement, Unified try/except/finally, try/finally Statement, raise Statement, assert Statement, catch multiple specific exceptions</li> <li>8. Demonstrate the concept of String-Based Exceptions, Class-Based Exceptions and Nesting Exception handlers.</li> <li>9. Demonstrate implementation of the Anonymous Function Lambda.</li> <li>10. Demonstrate implementation functional programming tools such as filter and reduce</li> <li>11. Demonstrate the Module Creation, Module usage.</li> <li>12. Demonstrate image insertion in python.</li> <li>13. Demonstrate use of DataFrame method and use of .csv files.</li> <li>14. Develop programs to learn GUI programming using Tkinter.</li> </ol>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ol style="list-style-type: none"> <li>1. Develop, document, and debug modular python programs to solve computational problems.</li> <li>2. Select a suitable programming construct and data structure for a situation.</li> <li>3. Use built-in strings, lists, sets, tuples and dictionary in applications.</li> <li>4. Define classes and use them in applications.</li> <li>5. Use files for I/O operations.</li> </ol>		

<i>Course Code: MCA-428 (A)</i>	<b>Data Science</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
<b>Course Objectives:</b> The objectives of the course are: <ol style="list-style-type: none"> <li>1) Define data science and explain its significance in various fields.</li> <li>2) Utilize data manipulation techniques and apply exploratory data analysis (EDA).</li> <li>3) Understand fundamental statistical concepts used in data science.</li> <li>4) Implement basic data visualization techniques and apply machine learning algorithms.</li> <li>5) Explain the importance of data ethics and privacy in data science projects.</li> </ol>		
<b>Unit 1:</b>	<b>[02]</b>	<b>Max Marks:05</b>
<b>Introduction to Data Science:</b> Definition and scope of data science, Data science applications in different domains, Introduction to data-driven decision making.		
<b>Unit-II</b>	<b>[04]</b>	<b>Max Marks:05</b>
<b>Data Collection and Preprocessing:</b> Data types and sources, Data acquisition and storage, Data cleaning and handling missing values.		
<b>Unit-III</b>	<b>[04]</b>	<b>Max Marks: 06</b>
<b>Exploratory Data Analysis (EDA):</b> Data summarization and descriptive statistics, Data visualization techniques, Identifying patterns and outliers in data.		
<b>Unit-IV</b>	<b>[04]</b>	<b>Max Marks: 06</b>
<b>Introduction to Statistics for Data Science:</b> Probability distributions, Hypothesis testing, Correlation and regression analysis.		
<b>Unit-V</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Data Visualization:</b> Introduction to data visualization tools (e.g., Matplotlib, Seaborn), Creating effective data visualizations, Design principles for data visualizations.		
<b>Unit-VI</b>	<b>[06]</b>	<b>Max Marks: 10</b>
<b>Introduction to Machine Learning:</b> Supervised vs. unsupervised learning, Model training and evaluation, Common machine learning algorithms (e.g., linear regression, decision trees).		
<b>Unit-VII</b>	<b>[02]</b>	<b>Max Marks: 04</b>
<b>Ethics and Privacy in Data Science:</b> Ethical considerations in data collection and analysis, Protecting sensitive information and ensuring privacy, Implications of biased data and algorithms.		
<b>Unit-VIII</b>	<b>[02]</b>	<b>Max Marks: 04</b>
<b>Data Science Project :</b> Forming teams and project planning, Data exploration and analysis, Presenting findings and conclusions.		
<b>References:</b> <ol style="list-style-type: none"> <li>1) John D. Kelleher and Brendan Tierney, Data Science., ISBN 9780262535434, The MIT Press.</li> <li>2) Wes Mc Kinney, Python for Data Analysis, O'Reilly.</li> <li>3) Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Introduction to Statistical Learning.</li> <li>4) Jojo Moolayil, Smarter Decisions: The Intersection of IoT and Data Science, PACKT, 2016.</li> <li>5) Cathy O'Neil and Rachel Schutt, Doing Data Science, O'Reilly, 2015.</li> <li>6) David Dietrich, Barry Heller, Beibei Yang, Data Science and Big data Analytics, EMC 2013</li> </ol>		

**Course Outcome:**

After completion of this course students shall be able to-

1. Identify and describe the methods and techniques commonly used in data science.
2. Demonstrate proficiency with the methods and techniques for obtaining, organizing, exploring, and analyzing data.
3. Recognize how data analysis, inferential statistics, modeling, machine learning, and statistical computing can be utilized in an integrated capacity.

<b>Course Code: MCA-428 (B)</b>	<b>Lab on Data Science</b>	<b>Clock Hours: 60</b> <b>Total Marks: 50</b> <b>Total Credits: 02</b>
---------------------------------	----------------------------	--

**Course Objectives:**

- 1) Learn Data Science concepts of Python and functioning of Python.
- 2) Understand Exploratory Data Science
- 3) Learn to program various analysis techniques

1. Write program for Creating and Manipulating Python Objects in Data Analysis
2. Write program to demonstrate Loops & Vectorization Missing Values.
3. Demonstrate Importing and exporting data.
4. Working with Pandas data frames
5. Develop python program for Basic plots using Matplotlib
6. Write program for Validating & Exploring Data Manipulations (Summarizing, Sorting, Sub-setting, Merging, joining)
7. Develop python program for Frequency distributions
8. Develop python program for Variability
9. Develop python program for Averages
10. Develop python program for Normal Curves
11. Write program to implement the following analysis techniques using Python
  - Statistical hypothesis generation and testing
  - Chi-Square test and t-Test
  - Analysis of variance
  - Correlation analysis
  - Maximum likelihood test
  - Regression analysis
  - Classification techniques
  - Clustering
  - Association rules analysis

**Course Outcome:**

After completion of this course students shall be able to-

1. Develop code using Python programming constructs for data analysis.
2. Manipulate data using python.
3. Write code for various data analysis techniques.



<i>Course Code: MCA-429 (A)</i>	<b>Machine Learning</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i> <i>Total Credits:02</i>
---------------------------------	-------------------------	---

***Course Objectives:***

The objectives of the course are:

- 1) The course gives understanding of fundamentals of Machine Learning such as its types, applications and other preliminaries.
- 2) Course gives fair idea about all important techniques of Machine Learning such as Classification, Regression and Clustering.
- 3) It also introduces Neural Network model and its applications to Machine Learning and touching Deep Learning.

<b>Unit-I</b>	<b>[04]</b>	<b>Max Marks:6</b>
---------------	-------------	--------------------

Introduction: Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation

<b>Unit-II</b>	<b>[04]</b>	<b>Max Marks:6</b>
----------------	-------------	--------------------

Regression(Linear, Lasso, Ridge), Decision trees, overfitting

<b>Unit-III</b>	<b>[05]</b>	<b>Max Marks: 8</b>
-----------------	-------------	---------------------

Instance based learning, Feature reduction, Collaborative filtering-based recommendation

<b>Unit-IV</b>	<b>[04]</b>	<b>Max Marks: 8</b>
----------------	-------------	---------------------

Probability and Bayes learning

<b>Unit-V</b>	<b>[04]</b>	<b>Max Marks:8</b>
---------------	-------------	--------------------

Logistic Regression, Support Vector Machine, Kernel function and Kernel SVM

<b>Unit-VI</b>	<b>[05]</b>	<b>Max Marks: 8</b>
----------------	-------------	---------------------

Neural network: Perceptron, multilayer network, backpropagation, introduction to deep neural network

<b>Unit-VII</b>	<b>[04]</b>	<b>Max Marks: 6</b>
-----------------	-------------	---------------------

Clustering: k-means, adaptive hierarchical clustering, Gaussian mixture model

***References:***

- 1) Tom Mitchell (1997). Machine Learning. First Edition, McGraw- Hill.
- 2) Ethem Alpaydin (2009). Introduction to Machine Learning Edition 2. The MIT Press.

***Course Outcome:***

After completion of this course students shall be able to-

1. Acquire in-depth knowledge of various facets of Machine Learning methods/techniques and algorithms.
2. Envisage practical application of Machine Learning to Business and Research Computational problems.
3. Use knowledge of Machine Learning for product/service development.

<b>Course Code: MCA-429 (B)</b>	<b>LAB on Machine Learning</b>	<b>Clock Hours: 60</b> <b>Total Marks: 50</b> <b>Total Credits: 02</b>
---------------------------------	--------------------------------	--

**Course Objectives:**

The objectives of the course are:

- 1) Make use of Data sets in implementing the machine learning algorithms
- 2) Implement various ML algorithms for Classification clustering, regression using a programming language of your choice preferably Python, R-Programming etc.
- 3) Implement the machine learning concepts and algorithms in any suitable language of choice.

1. Implement the Find-S Inductive Learning algorithm.
2. Implement the Candidate-Elimination Inductive Learning algorithm.
3. Write a program to implement Decision tree using Python/R/Programming language of your choice
4. Write program to calculate popular attribute selection measures (ASM) like Information Gain, Gain Ratio, and Gini Index etc. for decision tree.
5. Implement simple KNN using Euclidean distance in python.
6. Write a program to implement k-Nearest Neighbour algorithm to classify the iris dataset. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
7. Write a program to implement the naïve Bayesian classifier for a sample training dataset stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
8. Write a Program for Confusion Matrix and calculate Precision, Recall, F-Measure.
9. Write program for linear regression and find parameters like Sum of Squared Errors (SSE), Total Sum of Squares (SST),  $R^2$ , Adjusted  $R^2$  etc.
10. Implementing Agglomerative Clustering in python
11. Write a Program for Fuzzy c-means clustering in python.
12. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.
13. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

**References:**

- 1) Tom Mitchell (1997). Machine Learning. First Edition, McGraw- Hill.
- 2) Ethem Alpaydin (2009). Introduction to Machine Learning Edition 2. The MIT Press.
- 3) Dipanjan Sarkar, Raghav Bali, and Tushar Sharma, “ Practical Machine Learning with Python” A Problem-Solver’s Guide to Building Real-World Intelligent Systems, ISBN-13 (pbk): 978-1-4842-3206-4, Apress.

**Course Outcome:**

After completion of this course students shall be able to-

1. Understand the implementation procedures for the machine learning algorithms.
2. Design Java/Python programs for various Learning algorithms.

<i>Course Code: MCA-416</i>	<b>Research Methodology</b>	<i>Clock Hours: 60</i> <i>Total Marks: 100</i> <i>Total Credits:04</i>
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1)To study and understand the research issues &amp; challenges, research goals, scientific methods</li> <li>2)To study Sampling, External Validity, Levels of Measurement, Scaling and Qualitative Measures. Data Preparation, Descriptive Statistics and Correlation; and Inferential Statistics</li> <li>3)Reviewing Literature and Research Papers; Writing Research Papers, Thesis, Reports and Project Proposals Plagiarism and Copyrights.</li> </ol>		
<b>Unit-I</b>	<b>[08]</b>	<b>Max Marks:10</b>
Research Foundations: Research, Research Goals and Quality Research, Types of Research, Variables, Hypotheses and Data; Structure, Positivism and Post-Positivism; Scientific Methods, Reasoning and Arguments; Mathematical Methods of Proof and Research Fallacies.		
<b>Unit-II</b>	<b>[08]</b>	<b>Max Marks:15</b>
CS Research Context: Nature of Computer Science, Scientific Methods in Computer science, Types of Research in CS, Research Methods in Computer Science, Research Paradigms in CS, Grand Challenges for CS Research.		
<b>Unit-III</b>	<b>[10]</b>	<b>Max Marks: 12</b>
Measurements: Sampling, External Validity, Levels of Measurement, Scaling and Qualitative Measures. Research Design: Internal Validity, Types of Designs, Experimental Design, Probabilistic Equivalence, Hybrid Experimental Designs and Quasi-Experimental Design.		
<b>Unit-IV</b>	<b>[11]</b>	<b>Max Marks: 25</b>
Sampling: Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non-Response. Characteristics of a good sample. Probability Sample – Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage sampling. Determining size of the sample – Practical considerations in sampling and sample size.		
<b>Unit-V</b>	<b>[11]</b>	<b>Max Marks:20</b>
Data Analysis: Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association.		
<b>Unit-VI</b>	<b>[12]</b>	<b>Max Marks: 18</b>
Research Skills: Reviewing Literature and Research Papers; Writing Research Papers, Thesis, Reports and Project Proposals; Formatting, Appendices, Citation Formats and Style; General Conventions, Issues, Plagiarism and Copyrights.		
<b>References:</b> <ol style="list-style-type: none"> <li>1) Research Methodology: a step-by-step guide for beginners, Kumar, Pearson Education.</li> <li>2) Kothari C.K. (2004) 2/e, Research Methodology – Methods and Techniques (New Age International, New Delhi).</li> <li>3) Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd.</li> </ol>		

***Course Outcome:***

After completion of this course students shall be able to-

1. Understand the basic concepts of research and its methodologies, identify appropriate research topics, select and define appropriate research problems and parameters.
2. Prepare a project proposal.
3. Organize and conduct research in a more appropriate manner, writing research report and thesis.

## Semester-III

<i>Course Code: MCA-511</i>	<b>Design and Analysis of Algorithms</b>	<i>Clock Hours: 60</i> <i>Total Marks: 100</i>
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>To understand Basics of algorithms, design techniques and analyze the performance.</li> <li>To learn Searching and traversal algorithms for graphs.</li> <li>To understand Nondeterministic algorithms and NP class of problem.</li> </ul>		
<b>Unit-I</b>	<b>[06]</b>	<b>Max Marks:08</b>
What Is An Algorithm?, Algorithm Specification, reasons to study algorithms, Pseudocode Conventions, Recursive Algorithms with iterations and recursion, types of analysis, Asymptotic Notation, best, average and worst case analysis,		
<b>Unit-II</b>	<b>[06]</b>	<b>Max Marks:12</b>
Tree and Graph Representations, Binary Trees Basics, Heaps And Heap Sort, Sets And Disjoint Set Union And Find.		
<b>Unit-III</b>	<b>[12]</b>	<b>Max Marks:16</b>
<b>Divide And Conquer:</b> General Method, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Strassen's Matrix Multiplication.		
<b>Unit-IV</b>	<b>[08]</b>	<b>Max Marks:16</b>
<b>Greedy Method:</b> General Method, Optimal Storage on Tapes, Knapsack Problem, Huffman Code, Minimum-Cost Spanning Trees, Single-Source Shortest Paths.		
<b>Unit-V</b>	<b>[08]</b>	<b>Max Marks:14</b>
<b>Dynamic Programming:</b> General Method, All-Pair Shortest Path, Matrix Chain Multiplication, Longest Common Sub Sequence, 0/1knapsack, Flow Shop Scheduling		
<b>Unit-VI</b>	<b>[08]</b>	<b>Max Marks:10</b>
<b>Basic Search and Traversal Techniques:</b> Breadth First Search and Traversal, Depth First Search And Traversal, Spanning Trees.		
<b>Unit-VII</b>	<b>[06]</b>	<b>Max Marks:12</b>
<b>Backtracking:</b> General Method, Constrains, 8-Queens Problem Graph Coloring		
<b>Unit-VIII</b>	<b>[06]</b>	<b>Max Marks:12</b>
<b>NP-Hard and NP-Complete Problems:</b> Basic Concepts, Nondeterministic Algorithms, Polynomial Time, Polynomial-Time Verification, The Classes NP-Hard and NP-Complete, NP-Completeness and Reducibility, NP-Completeness Proofs, NP-Complete Problems		
<b>References:</b> <ul style="list-style-type: none"> <li>Horowitz E. and Sahni S. "Fundamentals of computer Algorithms" Galgotia publications.</li> <li>Horowitz E., Sahni S. and Rajshekar S, Computer Algorithms, Computer Science Press.</li> <li>S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani(2006), Algorithms. McGraw-Hill publications. ISBN 9780073523408</li> </ul>		

- Cormen, Leiserson and Rivest, Introduction to Algorithms,: Prentice Hall of India ISBN: 978- 81-203-4007-7

**Course Outcome:**

After completion of this course students shall be able to-

- Analyze the asymptotic performance of algorithms and write rigorous correctness proofs for algorithms.
- Design and analyze divide-and-conquer, greedy and dynamic-programming based algorithms.
- Model problems using backtracking, classify nondeterministic polynomial time algorithms.

<i>Course Code: MCA-512</i>	<b>Advanced Java Programming</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i>
<b>Course Objectives:</b> The objectives of the course are: <ul style="list-style-type: none"> <li>• To become familiar with the advanced features of Java Language as generic programming, collection framework.</li> <li>• To understand RMI technology and concept of reusable components using JavaBeans and EJB.</li> <li>• To understand Java Servlets, Java server Pages (JSP) technology and Strut &amp; Hibernate technology</li> </ul>		
<b>Unit-I</b>	<b>[04]</b>	<b>Max Marks:08</b>
<b>Generic programming:</b> Generic Class, Generic Method, Generic Interface; <b>The Collection framework:</b> HashSet, LinkedHashSet, Stack, LinkedList, ArrayList, HashMap. Vector Class, HashMap Class , Arrays Class		
<b>Unit-II</b>	<b>[06]</b>	<b>Max Marks:10</b>
<b>Servlets:</b> Concepts, Architecture, Servlet Container Writing Process, API, Life Cycle, Programming and deployment, Servlet and HTML Form, Session Management.		
<b>Unit-III</b>	<b>[08]</b>	<b>Max Marks:12</b>
<b>Java Server Pages:</b> Introduction, JSP Containers, Architecture, JSP and Servlets. Life Cycle of JSP Page, JSP Processing, Setting Up the JSP Environment, JSP Directives, JSP Action, JSP Implicit Objects, JSP Form Processing, JSP Session and Cookies Handling, JSP Session Tracking JSP Database Access, JSP Standard Tag Libraries, JSP Custom Tag, JSP Expression Language, JSP Exception Handling.		
<b>Unit-IV</b>	<b>[05]</b>	<b>Max Marks:10</b>
<b>Java Bean:</b> Concepts, Writing process, Applications, Properties and Events, <b>Enterprise JavaBeans:</b> Introduction, Specification, Architecture, Container, Types, Life cycle, Applications		
<b>Unit-V</b>	<b>[07]</b>	<b>Max Marks:10</b>
<b>MVC:</b> Introduction, Model-View-Controller Layers, Advantages and Disadvantages. <b>Strut:</b> Introduction, Simple Validation, Processing Business Logic, Basic Struts Tags, Configuring Struts, <b>Hibernate :</b> Overview of Hibernate, Hibernate Architecture, Hibernate Mapping Types, Hibernate Query Language.		
<b>References:</b> 1. Beginning Apache Struts From Novice to Professional, by Arnold Doray by Apress ISBN: 978-1-59059-604-3		

2. Professional Hibernate, by Eric Pugh, Joseph D. Gradecki by Wiley Publishing, Inc., ISBN: 0-7645-7677-1
4. Gary Cornell and Cay S. Horstmann, "Core Java Vol 1 and Vol 2", Sun Microsystems Press, Eight Edition.
5. Soumadip Ghosh, "Web Technology with Advanced Java", University Science Press, ISBN: 978-93-80856-78-0.

**Course Outcome:**

After completion of this course students shall be able to -

- Develop applications using advanced features of Java Language, RMI, JavaBeans and EJB.
- Use Java Servlets, JSP to develop web applications
- Use Struts and Hibernate in development of dynamic web applications.

<b>Course Code: MCA-513</b>	<b>LAB on Advanced Java Programming</b>	<b>Total Marks: 50</b>
<b>Course Objectives:</b>		
<ul style="list-style-type: none"> <li>• Study Step-by-Step procedure for building the project in java from ground up by using IDE.</li> <li>• Develop application using collection framework, RMI technology, JavaBeans and EJB</li> <li>• Develop Web Applications using advanced Java technology Servlets , JSP, Struts and Hibernate</li> </ul>		
<ul style="list-style-type: none"> <li>• Write java program(s) that demonstrates generic programming.</li> <li>• Write a Java program(s) that demonstrates the use of Collection Classes (Collection framework).</li> <li>• Write a Java program(s) that demonstrates Java Bean.</li> <li>• Write a Java program(s) that demonstrates EJB.</li> <li>• Write a Java program(s) that demonstrates use of Servlets.</li> <li>• Write a Java program(s) that demonstrates use of JSP technology.</li> <li>• Implement the dynamic web application(s) to demonstrate use of struts.</li> </ul>		
<b>Course Outcome:</b>		
After completion of this course students shall be able to-		
<ul style="list-style-type: none"> <li>• Step-by-Step procedure for building the project from ground up by using IDE.</li> <li>• Create dynamic web application to utilize the JavaBeans and EJBs reusable components</li> <li>• Create web application using servlets, JSP, Struts and Hibernate technologies.</li> </ul>		

<b>Course Code: MCA-514(A)</b>	<b>Natural Language Processing</b>	<b>Clock Hours: 30</b> <b>Total Marks: 50</b>
<b>Course Objectives:</b>		
<ul style="list-style-type: none"> <li>• The prime objective of this course is to introduce the students to the field of Language Computing and its applications ranging from classical era to modern context.</li> <li>• Course also aims to provide understanding of various NLP tasks and NLP abstractions such as Morphological analysis, POS tagging, concept of syntactic parsing, semantic analysis etc.</li> </ul>		

<ul style="list-style-type: none"> <li>Course provide knowledge of different approaches/algorithms for carrying out NLP tasks; it also discusses concepts of Language grammar and grammar representation in Computational Linguistics.</li> </ul>		
<b>Unit-I</b>	<b>[06]</b>	<b>Max Marks:08</b>
Introduction to NLP, brief history, NLP applications: Speech to Text(STT), Text to Speech(TTS), Story Understanding, NL Generation, QA system, Machine Translation, Text Summarization, Text classification, Sentiment Analysis, Grammar/Spell Checkers etc., challenges/Open Problems, NLP abstraction levels, Natural Language (NL) Characteristics and NL computing approaches/techniques and steps, NL tasks: Segmentation, Chunking, tagging, NER, Parsing, Word Sense Disambiguation, NL Generation, Web 2.0 Applications : Sentiment Analysis; Text Entailment; Cross Lingual Information Retrieval (CLIR).		
<b>Unit-II</b>	<b>[06]</b>	<b>Max Marks:08</b>
Text Processing Challenges, Overview of Language Scripts and their representation on Machines using Character Sets, Language, Corpus and Application Dependence issues, Segmentation: word level(Tokenization), Sentence level. Regular Expression and Automata Morphology, Types, Survey of English and Indian Languages Morphology, Morphological parsing FSA and FST, Porter stemmer, Rule based and Paradigm based Morphology, Human Morphological Processing, Machine Learning approaches.		
<b>Unit-III</b>	<b>[05]</b>	<b>Max Marks:10</b>
Word Classes ad Part-of-Speech tagging(POS), survey of POS tagsets, Rule based approaches (ENG TOWL), Stochastic approaches(Probabilistic, N-gram and HMM), TBL morphology, unknown word handling, evaluation metrics: Precision/Recall/F-measure, error analysis.		
<b>Unit-IV</b>	<b>[07]</b>	<b>Max Marks:12</b>
NL parsing basics, approaches: TopDown, BottomUp, Overview of Grammar Formalisms: constituency and dependency school, Grammar notations CFG, LFG, PCFG, LTAG, Feature-Unification, overview of English CFG, Indian Language Parsing in Paninian Karaka Theory, CFG parsing using Earley's and CYK algorithms, Probabilistic parsing, Dependency Parsing: Covington algorithm, MALT parser, MST parser.		
<b>Unit-V</b>	<b>[06]</b>	<b>Max Marks:12</b>
Concepts and issues in NL, Theories and approaches for Semantic Analysis, Meaning Representation, word similarity, Lexical Semantics, word senses and relationships, WordNet (English and IndoWordnet), Word Sense Disambiguation: Lesk Algorithm Walker's algorithm, Coreferences Resolution: Anaphora, Cataphora.		
<b>References:</b> <ul style="list-style-type: none"> <li>Indurkha, N., &amp; Damerau, F. J. (2010), Handbook of Natural Language Processing, 2nd Edition. New York: CRC Press Taylor and Francis Group, Boca Raton London, New York. ISBN-10: 1420085921, ISBN-13: 978-1420085921</li> <li>Martin, J. H., &amp; Jurafsky, D.(2013), Speech and Language Processing, Pearson Education India; 2 edition, ISBN-10: 9332518416, ISBN-13: 978-9332518414</li> <li>Manning, Christopher and Heinrich, Schutze(1999), Foundations of Statistical Natural Language Processing", MIT Press, ISBN-10: 0262133601, ISBN-13: 978-0262133609.</li> <li>Akshar Bharati, Chaitanya, V., Kulkarni, A., &amp; Sangal, R. (July 1997). Machine translation in</li> </ul>		



Stages (Vol. 10 no. 3). Mumbai: NCST, Mumbai.

- Bharati, A., Chaitanya, V., & Sangal, R. (1995). Natural Language Processing: A Paninian Perspective, New Delhi: Prentice Hall of India, ISBN 10: 8120309219, ISBN 13: 9788120309210.
- Steven Bird, Edward Loper (2016), Natural Language Processing With Python, Ed. 2, O'Reilly Media, ISBN 1491913428, 9781491913420

**Auxiliary Resources:**

Web Links

- <https://see.stanford.edu/Course/CS224N>
- <https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html>
- <https://stp.lingfil.uu.se/~nivre/docs/ACLslides.pdf>

**Video Links:**

- <http://www.nptelvideos.in/2012/11/natural-language-processing.html>
- <https://www.youtube.com/playlist?list=PL6397E4B26D00A269>

**Course Outcome:**

After completion of this course students shall be able to-

- Understand issues and challenges in Natural Language Processing and NLP applications and their relevance in the classical and modern context.
- Understand Computational techniques and approaches for solving NLP problems and develop modules for NLP tasks and tools.
- 3) Understand various grammar formalisms, which they can apply in different fields of study.

<b>Course Code: MCA-514(B)</b>	<b>Lab on Natural Language Processing</b>	<b>Total Marks: 50</b>
<b>Course Objectives:</b>		
The objectives of the course are:		
<ul style="list-style-type: none"><li>• Course provides knowledge of installation and use of NLTK in python.</li><li>• Course provides knowledge of implementation of text files processing operations and Regular Expressions in NLP</li><li>• Course provide knowledge of implementation of dependency parser, porter stemmer, Morphology, PoS Tagging</li></ul>		
<ul style="list-style-type: none"><li>• Install NLTK and perform basic preprocessing steps of NLP like tokenization, stemming, lemmatization, chunking etc using NLTK in python.</li><li>• Write a program to perform text files statistical operation like count number of lines in files, number of words in file.</li><li>• Working with PDF files in Python like Extracting text from PDF, Rotating PDF pages, Merging PDFs, Splitting PDF, Adding watermark to PDF pages</li><li>• Write program to count number of articles (a, an, the) in file.</li></ul>		

<ul style="list-style-type: none"> <li>• Write a program to perform tokenization and filtering stopwords in file.</li> <li>• Write a program which makes use of basics in regular expressions like /a*/, /a+/, /a? /, /[^A- Z]/, /[^Ss]/, etc.</li> <li>• Write a program for minimum edit distance algorithm.</li> <li>• Write a program for Understanding the morphology of a Marathi word. Take one or two suffixes of Marathi language and show the inflection on Gender, Number, Person, and Case.</li> <li>• Write a program to demonstrate use of porter stemmer in python.</li> <li>• Write a program to demonstrate use of dependency parser.</li> <li>• Write a program to demonstrate use of NP and VP chunker.</li> <li>• Write a program for Tagging Sentences which takes input as sentence and performs PoS Tagging.</li> <li>• Write a program for bigram formation from given list.</li> </ul>
<p><b>References:</b></p> <ul style="list-style-type: none"> <li>• Indurkha, N., &amp;Damerau, F. J. (2010), Handbook of Natural Language Processing, 2nd Edition. New York: CRC Press Taylor and Francis Group, Boca Raton London, New York. ISBN-10: 1420085921, ISBN-13: 978-1420085921</li> <li>• Martin, J. H., &amp;Jurafsky, D.(2013), Speech and Language Processing, Pearson Education India; 2 edition, ISBN-10: 9332518416, ISBN-13: 978-9332518414</li> <li>• Steven Bird, Edward Loper (2016),Natural Language Processing With Python, Ed. 2, O'Reilly Media,ISBN 1491913428, 9781491913420</li> </ul>
<p><b>Course Outcome:</b></p> <p>After completion of this course students shall be able to-</p> <ul style="list-style-type: none"> <li>• idea about installation and use of NLTK in python.</li> <li>• understanding of implementation of text files procesing operation and Regular Expressions in NLP</li> <li>• Knowledge of implementation of dependency parser, porter stemmer, Morphology, PoS Tagging and other NLP applications</li> </ul>

<b>Course Code: MCA-515 (A)</b>	<b>AI in Practice with Python</b>	<b>Clock Hours: 30</b> <b>Total Marks: 50</b>
<p><b>Course Objectives:</b></p> <p>The objectives of the course are:</p> <ul style="list-style-type: none"> <li>• How to make informed decisions about the type of algorithms you need to use and implementation of these algorithms for solving AI problems.</li> <li>• Use of artificial intelligence techniques for solving practical real-life problems.</li> <li>• To build real-world artificial intelligence applications.</li> </ul>		
<b>Unit-I</b>	<b>[02]</b>	<b>Max Marks:04</b>
<p><b>Machine Learning Pipelines</b></p> <p>What is a Machine Learning Pipeline? Problem Definition, Data Ingestion, Data Preparation,</p>		

Data Segregation, Model Training.		
<b>Unit-II</b>	<b>[04]</b>	<b>Max Marks:06</b>
<b>Feature Selection and Feature Engineering</b> Feature Selection, Feature Engineering, Outlier Management, One-hot Encoding, Log Transform, Scaling, Date Manipulation.		
<b>Unit-III</b>	<b>[06]</b>	<b>Max Marks:08</b>
<b>Classification and Regression Using Supervised Learning</b> Supervised versus Unsupervised Learning, What is Classification? Preprocessing Data, Label Encoding, Logistic Regression Classifiers, The Naive Bayes Classifier, Confusion Matrices, Support Vector Machines, What is Regression?, Building Single-Variable Regressor, Building Multivariable Regressor.		
<b>Unit-IV</b>	<b>[06]</b>	<b>Max Marks:08</b>
<b>Detecting Patterns with Unsupervised Learning</b> Unsupervised Learning, Clustering Data with K-Means Algorithm, Estimating Number of Clusters with the Mean Shift Algorithm, Estimating Quality of Clustering with Silhouette Scores.		
<b>Unit-V</b>	<b>[04]</b>	<b>Max Marks:06</b>
<b>Building Recommender Systems</b> Extracting the nearest Neighbors, Building K-Nearest Neighbors Classifier, Computing Similarity Scores, Finding Similar Users Using Collaborative Filtering, Case Study: Building Movie Recommender System.		
<b>Unit-VI</b>	<b>[03]</b>	<b>Max Marks:05</b>
<b>Building Speech Recognizer</b> Working with speech signals, Visualizing audio signals, Transforming audio signals into frequency domain, Generating audio signals, Synthesizing tones to generate music, Extracting speech features, Recognizing spoken words.		
<b>Unit-VII</b>	<b>[03]</b>	<b>Max Marks:05</b>
<b>Building NLP Tools</b> Installations of NLP packages, Tokenizing text data, Stemming, Lemmatization, Dividing text into chunks, Bag of words model, Building category predictor, Constructing gender Identifier, Building sentiment Analyzer.		
<b>Unit-VIII</b>	<b>[02]</b>	<b>Max Marks:03</b>
<b>Chatbots</b> Chatbots today, Concepts, A well-architected chatbot, Platforms, Creating a chatbot,		
<b>References:</b> <ul style="list-style-type: none"> <li>• Python Machine Learning, Sebastian Raschka, Vahid Mirjalili, Packt Publishing, ISBN: 978-1- 78995-575-0</li> <li>• AI Crash Course, Hadelin de Ponteves, Packt Publishing, ISBN: 978-1-83864-535-9</li> </ul>		

**Course Outcome:**

After completion of this course students shall be able to-

- Develop practical AI applications with solid understanding of many new AI techniques.
- Implement more complex AI algorithms using Python.
- Use AI algorithms to create new real world AI applications.

<b>Course Code: MCA-515 (B)</b>	<b>LAB on AI Practice using Python</b>	<b>Total Marks: 50</b>
---------------------------------	--	------------------------

**Course Objectives:**

- To explore most common artificial intelligence (AI) use cases.
  - To implement various new artificial intelligence techniques.
  - To create real-world AI application/s using above AI technique/s.
- 
- Installation of Python on Windows/Ubuntu, Installing Packages, Loading data.
  - Data Preparation using techniques like Data Cleansing, Filtration, Aggregation etc
  - Handling missing values, Feature Scaling, Inconsistent values in the given dataset.
  - Feature selection using techniques like univariate selection correlation heatmaps, Wrapper-based methods, Filter-based methods.
  - Feature engineering using techniques like Outlier management, One-hot encoding, Log transform.
  - Implement Logistic regression classifier.
  - Implement Naïve Bayes classifier.
  - Use confusion matrixes to describe the performance of a classifier.
  - Implement classifier using Support Vector Machines.
  - Build a decision tree classifier and evaluate performance of a classifier by printing classification report.
  - Build random forest and extremely random forest classifiers and analyze the output.
  - Implement K-Means algorithm for clustering.
  - Build K-nearest classifier.
  - Visualizing audio signals.
  - Transform audio signals to the frequency domain.
  - Generate audio signals.
  - Installation of NLTK and tokenizing text data.
  - Converting words to their base forms using stemming and lemmatization.
  - Extracting the frequency of terms using Bag of Words model.

**Course Outcome:**

After completion of this course students shall be able to-

- Use most common artificial intelligence (AI) use cases in developing AI applications.
- Apply various new artificial intelligence techniques in developing AI applications.
- Create real-world AI application/s using above AI technique/s.

<i>Course Code: MCA-516(A)</i>	<b>High Performance Computing Paradigms and Applications</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i>
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>To introduce the concepts of parallel and distributed computing.</li> <li>To understand high-performance architectures and parallel programming models.</li> <li>To develop basic skills in parallel algorithm design and implementation.</li> <li>To use industry-standard HPC tools and libraries.</li> </ul>		
<b>Unit-I Introduction to HPC and Shared Memory Systems</b>	<b>[04]</b>	<b>Max Marks:06</b>
<ul style="list-style-type: none"> <li>Evolution and need for HPC</li> <li>Flynn's Taxonomy: SISD, SIMD, MISD, MIMD</li> <li>Shared vs Distributed Memory Architectures</li> <li>Concepts: Parallelism, Concurrency, Speedup, Scalability</li> </ul>		
<b>Unit-II Fundamentals of Parallel Programming</b>	<b>[04]</b>	<b>Max Marks:06</b>
<ul style="list-style-type: none"> <li>Parallel Programming Models: Thread-based vs Process-based</li> <li>Task vs Data Parallelism</li> <li>Amdahl's Law and Scalability Limits</li> <li>Compiler Directives, Threads, and Processes</li> </ul>		
<b>Unit-III Introduction to OpenMP</b>	<b>[06]</b>	<b>Max Marks:08</b>
<ul style="list-style-type: none"> <li>OpenMP Execution Model and Environment</li> <li>Parallel Region, Threads, and Fork-Join Model</li> <li>Basic Directives: parallel, for, sections</li> <li>Compiler setup and sample programs</li> </ul>		
<b>Unit-IV OpenMP Synchronization and Work Sharing</b>	<b>[06]</b>	<b>Max Marks:08</b>
<ul style="list-style-type: none"> <li>Work Sharing Constructs: for, sections, single, master</li> <li>Synchronization: critical, atomic, barrier, ordered</li> <li>Variable Scope: shared, private, firstprivate, lastprivate</li> <li>nowait, flush, threadprivate</li> </ul>		
<b>Unit-V Performance Tuning and Scheduling in OpenMP</b>	<b>[04]</b>	<b>Max Marks:08</b>
<ul style="list-style-type: none"> <li>Scheduling policies: static, dynamic, guided, auto</li> <li>Use of reduction clause</li> <li>Timing functions: omp_get_wtime()</li> <li>Profiling and Optimization strategies</li> <li>Avoiding race conditions and false sharing</li> </ul>		
<b>Unit- VI Distributed Memory Programming using MPI</b>	<b>[04]</b>	<b>Max Marks:10</b>
<ul style="list-style-type: none"> <li>MPI Environment and Communication</li> <li>Point-to-Point and Collective Communication</li> <li>Scatter, Gather, Broadcast</li> </ul>		

Sample MPI Programs		
<b>Unit-VII Applications and Trends in HPC</b>	<b>[02]</b>	Max Marks:04
<ul style="list-style-type: none"> <li>Real-world Applications (Weather Forecasting, Bioinformatics, ML, etc.)</li> <li>GPUs and CUDA (Introductory Overview)</li> <li>HPC in Cloud &amp; Exascale Computing</li> </ul>		
<b>References</b> <ol style="list-style-type: none"> <li><b>Ananth Grama et al.</b> – <i>Introduction to Parallel Computing</i></li> <li><b>Barbara Chapman et al.</b> – <i>Using OpenMP</i></li> <li><b>Peter Pacheco</b> – <i>Parallel Programming with MPI</i></li> <li>Online Resources: <ul style="list-style-type: none"> <li><a href="https://openmp.org">https://openmp.org</a></li> <li><a href="https://mpi4py.readthedocs.io">https://mpi4py.readthedocs.io</a></li> </ul> </li> </ol>		
<b>Course Outcome:</b> On completion of this course, learners will be able to: <ol style="list-style-type: none"> <li>Understand the architecture and models of HPC systems.</li> <li>Identify and analyze performance bottlenecks.</li> <li>Implement and evaluate basic parallel programs.</li> </ol> Utilize MPI/OpenMP for parallel computing.		

<i>Course Code: MCA-516 (B)</i>	<b>Lab on High Performance Computing</b>	<i>Total Marks: 50</i>
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>To introduce the concepts of parallel and distributed computing.</li> <li>To understand high-performance architectures and parallel programming models.</li> <li>To develop basic skills in parallel algorithm design and implementation.</li> <li>To use industry-standard HPC tools and libraries</li> </ul>		
<ul style="list-style-type: none"> <li>Write a basic OpenMP program where each thread prints its thread ID.( Parallel Hello World)</li> <li>Implement matrix addition with and without OpenMP. (Matrix Addition using OpenMP)</li> <li>Use OpenMP #pragma omp parallel for to parallelize nested loops. (Matrix Multiplication with Parallel for)</li> <li>Perform summation of array elements using OpenMP. (Parallel Summation with Reduction Clause)</li> <li>Parallelize the search for maximum/minimum values using reduction. (Finding Maximum/Minimum in an Array)</li> <li>Implement a parallel version of bubble sort (though not optimal for HPC, good for concept).( Parallel Bubble Sort)</li> <li>Compare run-times and calculate speedup for sequential and OpenMP versions of a function (e.g., matrix multiplication or sum of array).</li> <li>Explore limitations of OpenMP in recursive algorithms. (Fibonacci Numbers: Recursive vs Iterative Parallelism)</li> <li>Parallelize the process of finding all prime numbers up to N. (Parallel Prime Number Finder)</li> </ul>		

- Implement integration using a for loop and apply OpenMP for parallelizing it. (Numerical Integration using Trapezoidal Rule)
- Use #pragma omp sections to run different functions in parallel (e.g., I/O and computation simultaneously). (OpenMP Sections Directive)
- Estimate the value of Pi using Monte Carlo method.
- Generate a histogram from a dataset (e.g., grayscale image pixel values).
- Demonstrate static, dynamic, and guided scheduling. (Loop Scheduling Experiments)

**Course Outcome:**

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

- 1) Understand the architecture and models of HPC systems.
- 2) Identify and analyze performance bottlenecks.
- 3) Implement and evaluate basic parallel programs.
- 4) Utilize MPI/OpenMP for parallel computing.

<i>Course Code: MCA-517(A)</i>	<b>Mobile Application Development (Android Programming)</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i>
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>• Understand basics of mobile application development and get introduced Android platform and its architecture.</li> <li>• To learn activity creation and Android UI designing.</li> <li>• To be familiarized with Intent, Broadcast receivers and Internet services, SQLite Database and content providers; to integrate multimedia, camera and Location based services in Android Application.</li> </ul>		
<b>Unit- I</b>	<b>[05]</b>	<b>Max Marks:04</b>
<b>Mobile Application Development:</b> Introduction to handheld devices, Device Applications Vs Desktop and Web application, overview of application development platform: Android, Comparison of Android with other Mobile OS.		
<b>Unit-II</b>	<b>[05]</b>	<b>Max Marks:08</b>
<b>Hello, Android and Installations:</b> background, what is android and what is not?, Open Mobile Development Platform, Native Android Applications, Android SDK Features, Introducing the Development Framework, Installation, Emulator.		
<b>Unit-III</b>	<b>[07]</b>	<b>Max Marks:12</b>
<b>Activities, UI Design, and Navigation:</b> Introducing the Application Manifest. Using the Manifest Editor. The Android Application Life Cycle. Understanding Application Priority and Process States. Externalizing Resources. A Closer Look at Android Activities. Fundamental Android UI Design. Introducing Views. Introducing Layouts and fragments, Using Adapters, Creating New Views.		
<b>Unit-IV</b>	<b>[05]</b>	<b>Max Marks:12</b>
<b>Intents, Broadcast Receivers, working with the Background:</b> Using Intents to Launch Activities, Explicit Intents, Implicit Intents, Pending Intents, Intent Filters and Broadcast Receivers, Using Toast, Introducing Notifications. Introducing Dialogs and Action Bars.		

<b>Unit-V</b>	<b>[08]</b>	<b>Max Marks:06</b>
<b>Preferences, Databases and Content Providers:</b> Introducing Android Databases, Shared Preferences: Creating, Saving, Retrieving, SQLite Databases, Content Providers, Content Values and Cursors, data manipulation using SQLite.		
<b>References:</b> <ul style="list-style-type: none"> <li>• Reto Meier. Professional Android Application Development, Wrox Publications ISBN: 978-0- 470-34471-2.</li> <li>• Rick Rogers, John Lombardo, Zigurd Mednieks, G. Blake Meike. Android Application Development: Programming with the Google SDK. O'Reilly ISBN 10: 0596521472 / ISBN 13: 9780596521479.</li> <li>• Auxiliary Resources: <a href="https://developer.android.com/index.html">https://developer.android.com/index.html</a></li> </ul>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ul style="list-style-type: none"> <li>• Compare android with other smartphone OS and desktop OS; Able to understand software stack of android OS.</li> <li>• Understand Activity lifecycle, UI management, use Intent, Broadcast receivers and Internet services.</li> <li>• Effectively use SQLite Database and content providers, multimedia, camera and Location based services in Android Application.</li> </ul>		
<b>Course Code: MCA-517 (B)</b>	<b>Lab on Mobile</b> <b>Application Development (Android Programming) (2) (P)</b>	<b>Total Marks: 50</b>
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>• To gain knowledge of installing Android Studio and Cross Platform Integrated Development Environment;</li> <li>• To learn designing of User Interface and Layouts for Android App, intents to broadcast data within and between Applications.</li> <li>• To use Content providers and Handle Databases using SQLite. .</li> </ul>		
<ul style="list-style-type: none"> <li>• Assignments:</li> <li>• Demonstrate string manipulation by displaying at the middle of the screen in the red color with white background with change in fonts &amp; styles of text.</li> <li>• Write a program to populate resources (res&gt;&gt;value folder). Show resource on changing selection of the resources.</li> <li>• Write a program to create UI with one screen having radio button of the types of cars. On selecting any car name, next screen should show car details.</li> <li>• Write a program for android application to demonstrate android life cycle stages.</li> <li>• Create the application that will change color of screen based on selected option from the menu.</li> <li>• Write an android application that takes input from user and shows messages on screen.</li> <li>• Create foreground application that will display toast (Message) on specific interval time.</li> <li>• Demonstrate use of intents for any 3 native intents.</li> <li>• Create the android application that will read phonebook contact using content providers and</li> </ul>		



<p>display in list on selecting specific contact makes a call to selected contact.</p> <ul style="list-style-type: none"> <li>• Develop android application to take a picture using native application.</li> <li>• Use fragments to develop UI. Demonstrate use of fragments.</li> <li>• Demonstrate use of shared preferences.</li> <li>• Write code that will call maps using android application.</li> <li>• Develop application for database manipulation.</li> <li>• <b>Mini project:</b> Develop an app in android.</li> </ul>
<p><b>Course Outcome:</b></p> <p>After completion of this course students shall be able to-</p> <ul style="list-style-type: none"> <li>• 1) Design and Implement User Interfaces and Layouts of Android App; Use Intents for activity and broadcasting data in Android App.</li> <li>• 2) Design and Implement Database Application and Content Providers.</li> <li>• 3) Develop Android App with Security features.</li> </ul>

<i>Course Code: MCA-518 (A)</i>	<b>Microsoft .Net Technologies</b>	<i>Clock Hours:30</i> <i>Total Marks: 50</i>
<p><b>Course Objectives:</b></p> <ul style="list-style-type: none"> <li>• Provide a foundational understanding of the .NET framework and its role in web development.</li> <li>• Equip students with practical skills in designing web pages using server controls.</li> <li>• Enable students to connect web applications to databases using .NET.</li> <li>• Teach students to generate and customize reports within .NET applications.</li> </ul>		
<b>Unit-I: Introduction to .NET &amp; Web Server Controls</b>	<b>[05]</b>	<b>Max Marks:10</b>
<p>Overview of .NET framework, CLR, CTS, CLS, ASP.NET application architecture, Creating web forms, Using standard web server controls (Label, TextBox, Button, CheckBox, RadioButton, etc.), Validation controls (RequiredFieldValidator, RangeValidator, etc.), Navigation controls (Menu, TreeView, SiteMapPath), Designing and structuring web pages</p>		
<b>Unit-II: CSS, Themes, and Master Pages</b>	<b>[05]</b>	<b>Max Marks:10</b>
<p>CSS: Syntax, selectors, and integration with ASP.NET pages, Creating and applying themes in ASP.NET, Skins: Creating and using .skin files, Master pages: Purpose, creation, and usage, Content pages and ContentPlaceHolder controls, Layout management using master pages, Consistent UI design with themes and CSS, Applying themes and CSS to master and content pages</p>		
<b>Unit-III: Database Connectivity in .NET</b>	<b>[10]</b>	<b>Max Marks:15</b>
<p>ADO.NET components and architecture, Establishing connections (SqlConnection, OleDbConnection), Executing commands (SqlCommand, OleDbCommand), Reading data (SqlDataReader, DataSet, DataAdapter), Data binding to controls (GridView, DataList),</p>		

Exception handling in database operations		
<b>Unit-IV: Creating Reports in .NET</b>	<b>[10]</b>	<b>Max Marks:15</b>
Introduction to reporting in .NET, Crystal Reports and ReportViewer controls, Designing reports: layout, formatting, grouping, sorting, Connecting reports to data sources, Embedding and displaying reports in web applications, Exporting reports to PDF, Excel, etc., Printing reports		
<b>References:</b> <ul style="list-style-type: none"> <li>• ASP.NET with C# by Example, Tata McGraw-Hill Education by Vipul Patel</li> <li>• Professional ASP.NET 2.0 Design: CSS, Themes, and Master Pages by Dino Esposito, Wiley</li> <li>• "Professional ASP.NET 4.5 in C# and VB" by Jason N. Gaylord et al.</li> <li>• "Beginning ASP.NET 4.5: in C# and VB" by Imar Spaanjaars</li> </ul>		
<b>Course Outcome:</b> After completion of this course students shall be able to- <ul style="list-style-type: none"> <li>• Explain the architecture and features of the .NET framework.</li> <li>• Design and implement web pages using various web server controls.</li> <li>• Establish and manage database connectivity in .NET applications.</li> <li>• Create, format, and present reports using .NET reporting tools.</li> </ul>		

<b>Course Code: MCA-518 (B)</b>	<b>Lab on Microsoft .Net Technologies</b>	<b>Total Marks: 50</b>
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>• 1) Students will understand Web Sites / Web applications, basics of Web hosting and working of IIS web server.</li> <li>• 2) Set up a programming environment for ASP.Net programs, configure an ASP.Net application, creating standard .net controls based and data driven web application using ASP.Net; Maintain session and controls related information for user used in multi-user web applications.</li> <li>• 3) Understand the fundamentals of developing modular application by using object-oriented methodologies.</li> </ul>		
<ul style="list-style-type: none"> <li>• Demonstrate concept of postback and viewstate using web form server controls of ASP.NET</li> <li>• Demonstrate various Web form server controls using sample data entry screen form for registering for a service on website. Also use validation controls to validate input data.</li> <li>• Demonstrate DropDown List box, CheckButtonList, RadioButtonList controls.</li> <li>• Demonstrate Databinding using Hashtable, ArrayList, DataTable data sources.</li> <li>• Demonstrate Repeater control with the help of various templates.</li> <li>• Demonstrate paging, sorting, filtering of data in asp:DataGrid/DataGridView.</li> <li>• Demonstrate editing process in DataGrid and DataList controls. Make use of necessary</li> </ul>		

templates for proper visual appearance. <ul style="list-style-type: none"> <li>• Create sample website for demonstrating use of Profiles/Themes using skin files.</li> <li>• Demonstrate Master Pages and website navigation controls(sitemap path, treeview, menu) using SiteMap file.</li> <li>• Demonstrate Properties of website navigation controls.</li> <li>• Demonstrate creation of simple/complex DataReader/DataSet Objects.</li> <li>• Demonstrate editing in DataTable objects.</li> </ul>
<b>Course Outcome:</b> After completion of this course students shall be able to- <ul style="list-style-type: none"> <li>• Use most common artificial intelligence (AI) use cases in developing AI applications.</li> <li>• Apply various new artificial intelligence techniques in developing AI applications.</li> <li>• Create real-world AI application/s using above AI technique/s.</li> </ul>

<i>Course Code: MCA-519(A)</i>	<b>Ruby on Rails</b>	<i>Clock Hours: 30</i> <i>Total Marks: 50</i>
<b>Course Objectives:</b> <ul style="list-style-type: none"> <li>• Learn and understand Ruby and Rails.</li> <li>• Learn to develop programs using lexical and syntactic structure of Ruby programs.</li> <li>• Learn the use of Ruby TK (GUI for Ruby).</li> <li>• Design web applications using Rails framework.</li> </ul>		
<b>Unit-I</b>	<b>[02]</b>	<b>Max Marks:06</b>
<b>Ruby Installation and Basics:</b> Introduction to Ruby, Ruby naming convention, Interactive Ruby (IRB) & “ri” (Ruby Interactive) command-line tools, Ruby object, Ruby types: String, Hash, Symbol, Ruby class, Inheritance, Ways of creating Ruby object, Ruby methods, Methods Basics, Methods Advanced: Arguments, Visibility, Method with a! (bang), Modules, Control structures, Exception handling, Ruby operators, Regular expression		
<b>Unit-II</b>	<b>[06]</b>	<b>Max Marks:06</b>
<b>Ruby core:</b> Basics of block, How does a block look like? Block passing and execution, Proc, & (Ampersand), lambda, Closure, What is and Why Meta-programming?, Ruby language characteristics (that make it a great metaprogramming language), Object#respond_to?, Object#send, Dynamic typing (and Duck typing), missing_method, define_method		
<b>Unit-III</b>	<b>[04]</b>	<b>Max Marks:08</b>
<b>Ruby I/O:</b> File I/O, File inquiries, Directories, Navigation through Directories		
<b>Unit-IV</b>	<b>[10]</b>	<b>Max Marks:16</b>
<b>Ruby Advanced:</b> Ruby OOPs concepts, Database Access, Web Application without framework, Sending email Ruby TK (GUI for Ruby), Ruby – Multithreading Built in Functions, Variables, Constants, Ruby associated tools, Ruby - XML, XSLT and XPath		
<b>Unit-VI</b>	<b>[08]</b>	<b>Max Marks:14</b>
<b>Rails Basics:</b> What is and Why Ruby on Rails? Building HelloWorld Rails application step by		

step, App directory structure (MVC), Environment, Rake, Gems, Generators, Migration, Console, Bundle, scaffolding, ORM (ActiveRecord), Action controller basics, Action Views, Helpers, Authentication. **Application:** Rails Associations, JSON, APIs, and Oauth, Debugging Rails Application, Action Mailer, Rails Command lines/rails console, Securing Rails Application, Active Support, Rails Internationalization

**Unit Test:** Rspec

**Deployment:** Heroku deployment

**References:**

- Yukihiro Matsumoto (2008), The Ruby Programming Language, Shroff; First edition, 2008, ISBN- 10: 8184044925, ISBN-13: 978-8184044928
- Michael Fitzgerald, Learning Ruby, Published by O'Reilly Media, Inc., May 2007, IS BN-10: 8184043341, ISBN-13: 978-8184043341
- Rails AntiPatterns, Wesley Professional Ruby Series, 1st edition, 2010, ISBN-10: 0321604814, ISBN-13: 978-0321604811
- Adam Gamble, Cloves Carneiro, Jr. Rida Al Barazi (2007), Beginning Rails4, Apress, 3rd edition, 2013 ISBN-13 (pbk): 978-1-4302-6034-9| ISBN-13 (electronic): 978-1-4302-6035-6

**Course Outcome:**

After completion of this course students shall be able to-

- Understand Ruby Programming language with lexical and syntactic structure of Ruby programs, Datatypes and Objects, Expressions and Operators, Statements and Control Structures, Methods, procs, lambdas, and closures, Classes and modules, Reflection and Metaprogramming.
- Use Ruby TK (GUI for Ruby).
- Design web applications using Rails framework.

<i>Course Code: MCA-519 (B)</i>	<b>Lab on Ruby on Rails</b>	<i>Total Marks: 50</i>
<b>Course Objectives:</b>		
<ul style="list-style-type: none"> <li>• 1) Install Ruby on Rails</li> <li>• 2) Write programs in ruby.</li> <li>• 3) Develop applications using rails framework.</li> </ul>		
<ul style="list-style-type: none"> <li>• 1. Demonstrate a Ruby Basic program which manipulate Hash, Array, Strings. Any five methods of each container. Also use !(bang) operator.</li> <li>• 2. Write Ruby program which accepts user input and process it then print the result. Like radius.rb is a file which accept input as float and returns a calculated Area of circle  cirltom@laptop:~/courses/ruby/asst\$ ruby radius.rb  Enter the radius: 2  Area is: 12.5663708</li> <li>• 3. Write a ruby program which prompts for and reads one line of input. It then echos the line, then prints it repeatedly, each time removing every second character. It continues until no more characters can be removed. Treat all characters alike; no special treatment</li> </ul>		

for spaces or punctuation.

- 4. Demonstrate Inheritance in Ruby by building a superclass called Bird from which our Duck, Goose, and Owl classes will derive their functionality.  
([http://www.gotealeaf.com/books/oo\\_ruby/read/inheritance](http://www.gotealeaf.com/books/oo_ruby/read/inheritance))
- 5. Demonstrate a Ruby programs which uses loops like, each, times, do loop, etc. With having use of operators & exceptions which cause to break loop like divide by zero, etc.
- 6. Write a Ruby script which demonstrate use of blocks, lambda & proc.
- 7. Write a Ruby program which show duck typing, uses respond\_to? Method.
- 8. Write a Ruby program which define dynamic methods and method will return something also use missing\_method. It should return some result to console if some method is missing.
- 9. Create a Basic Ruby on Rails web application which print “Hello World on web browser”
- 10. Create a Ruby on Rails web application which shows having Post Section. In which user can Insert, Edit, Delete Post, using scaffolding.
- 11. Create a Ruby on Rails web application with Post Model uses various type of server validation.
- 12. Create a Ruby on Rails web application which shows having Post Section. In which user can Insert, Edit, Delete Post, using scaffolding, using mysql database.
- 13. Create a Ruby on Rails web application using mysql database without scaffold. which shows having Post Section. In which user can Insert, Edit, Delete Post. Post have multiple comments, comments can also Insert, Edit, Delete with nested routes like “/posts/2/comments”
- 14. Create a Ruby on Rails web application using mysql database. Post is always belongs to user and user has many posts. Without login user can't Insert, Delete or Edit Post, can only show post using devise gem.

***Course Outcome:***

After completion of this course students shall be able to-

- 1) Develop program using syntactic structure in ruby.
- 2) Build program using APIs of Ruby Programming Language.
- 3) Design web applications using Rails framework.

<i>Course Code: MCA-520</i>	<b>Research Project</b>	<i>Total Marks: 150 Clock Hours : 180</i>
<p><b>Course Objectives:</b></p> <p>A project is an in-depth study of an issue or topic in computational sciences, information technology or computing. It may be in the form of a small-scale research study, a case study, hardware/software implementation of research paper or innovative software solution.</p> <p>In this course it is expected that the student will complete the Research Project initiated in the previous semester (MCA-416) by identifying the suitable methodologies/techniques, Analyze data and synthesize research findings, to solve the problem. The student will report research findings in written and verbal forms.</p>		
<p><b>Guidelines for the Research Project :</b></p> <ol style="list-style-type: none"> <li>1) The students should continue the work initiated in the previous semester (MCA-416) by identifying the suitable methodologies/techniques, Analyze data and synthesize research findings, to solve the problem.</li> <li>2) The student should clearly mention the need of project, database(s), files required for the project, software used for the project, reasons for selection of that software, inputs required, outputs produced etc.</li> <li>3) The student has to prepare a report based on the actual work undertaken at the specific selected enterprise/ organization or sub system and get it certified by the concerned teacher that the Project report has been satisfactorily completed and submit TWO typed copies (with Spiral Binding) of the same to the Head / Director of the institute /Principal of the college.</li> <li>4) One copy of the report submitted by the student shall be maintained/kept with respective recognized Research Laboratory of the College/ Institute. 5) No student will be permitted to appear for Viva-Voce examinations, unless and until the project report is submitted within the stipulated time.</li> </ol> <p><b>Presentation of your project report</b></p> <p><b>Template:</b></p> <p>The contents of the Dissertation shall be arranged in the following order:</p> <ul style="list-style-type: none"> <li>• Cover Page</li> <li>• Inside Title Page</li> <li>• Certificate signed by the Supervisor and HOD</li> <li>• Declaration signed by the Candidate</li> <li>• Acknowledgements</li> <li>• Abstract</li> </ul>		

- Table of Contents
  - List of Figures
  - List of Tables
  - Abbreviations/ Notations/ Nomenclature (if any)
  - Text of the Report
- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Methodology
- Chapter 4: Results and Discussions
- Chapter 5: Conclusion and Recommendations
- References
  - Appendices
  - Non-paper materials (if any)

***Course Outcomes:***

On completion of this course, students will have the knowledge, skills and understanding to enable them to:

- Apply critical thinking skills.
- Apply foundational research skills to address a research question.
- Demonstrate planning, time and change management skills.
- Analyze data and synthesize research findings.
- Report research findings in written and verbal forms.
- Use research findings to advance education theory and practice.
- Undertake research independently.