KAVAYITRI BAHINABAI CHAUDHARI NORTH MAHARASHTRA UNIVERSITY, JALGAON

||अंतरी पेटवू ज्ञानज्योत||

1990
'A' Grade
NAAC Re-Accredited
(3rd Cycle)

MASTER OF COMPUTER APPLICATIONS
(MCA)

(at University Campus under Academic Flexibility w.e.f. 2020-21)

(w.e.f. Academic Year 2020-21)

Summary of Distribution of Credits under CBCS Scheme for M.C.A. at School of Computer Sciences

[at University Campus under Academic Flexibility w.e.f. 2020-21]

Sr. No.	Type of course	Sem I	Sem II	Sem III	Sem IV
01	Core	22	22	16	-
02	Skill based	06	06	06	-
03	School Elective	-	-	06	-
04	Project	-	-	-	12
05	Audit	02	02	02	-
06	Total Credits	30	30	30	12

Subject Type	Core	Skill based	School Elective	Project	Audit	Total
Credits	60	18	06	12	06	102

Total Credits = 102

Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon

School of Computer Sciences
M. C. A. (w. e. f. A. Y. 2020 -2021)
Course credit scheme

Semester	(A) Core Courses			(B) Skill Based / Elective Course			(C) Audit Course (No weightage in CGPA)			Total Credits (A+B+C)
	No. of Courses	Credits (T+P)	Total Credits	No. of Courses	Credits (T+P)	Total Credits	No. of Courses	Credits (Pract.)	Total Credits	
I	4	16 + 6	**22**	1	4 + 2	6	1	2	2	30
II	4	16 + 6	**22**	1	4 +2	6	1	2	2	30
III	4	16 + 6	**22**	1	4 +2	6	1	2	2	30
IV	1	12	**12**	-	-	-	-	-	-	12
Total Credits	**72**			**24**			**6**			**102**

(T, Theory; P, Practical)

Structure of Curriculum

		First Year				Second Year				Total Credit Value
		Semester I		Semester II		Semester III		Semester IV		
		Credit	Course	Credit	Course	Credit	Course	Credit	Course	
	Prerequisite and Core Courses									
(A)	Theory	16	4	16	4	16	4	-	-	**44**
	Practical	6	3	6	3	6	3	-	-	**16**
	Project(Industrial Training)	-	-	-	-	-	-	12	1	**12**
(B)	**Skill Based / Subject Elective Courses**									
	Theory	4	1	4	1	4	1	-	-	**16**
	Practical	2	1	2	1	2	1	-	-	**08**
(C)	**Audit Course (No weightage in CGPA calculations)**									
1	Practicing Cleanliness	2	1	-	-	-	-	-	-	**2**
2	Personality & and Cultural Development Related Course	-	-	2	1	-	-	-	-	**2**
3	Technology Related + Value Added Course	-	-	-	-	2	1	-	-	**2**
4	Professional /Social + Value added course	-	-	-	-	-	-	-	-	
	Total Credit Value	30	10	30	10	30	10			**102**

List of Audit Courses (Select any ONE course of Choice from Semester II and Semester III)					
Semester I (Compulsory)		Semester II (Choose One) Personality and Cultural Development		Semester III (Choose One) Technology + Value Added Course	
	Course Title		Course Title		Course Title
AC-101	Practicing Cleanliness	AC-201 (A)	Soft Skills	AC-301(A)	Computer Skills
		AC-201 (B)	Sport Activities	AC-301(B)	Cyber Security
		AC-201 (C)	Yoga	AC-301(C)	Linux (Spoken Tutorial Course)
		AC-201 (D)	Music	AC-301(D)	Advance C++ (Spoken Tutorial Course)

Semester I

Total Credit for Semester I: 30 (T = Theory: 16; P = Practical: 6; Skill Based: 6; Audit Course:2)

	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination					Credits
						Internal		External		Total	
			T	P	Total	T	P	T	P		
CA-101	Core	Database Management System (DBMS)	04	-	04	40	-	60	-	100	04
CA-102	Core	Operating Systems	04	-	04	40	-	60	-	100	04
CA-103	Core	Fundamentals of Artificial Intelligence	04	-	04	40	-	60	-	100	04
CA-104(A)	Core	Computer Programming and Problem Solving	04	-	04	40	-	60	-	100	04
CA-104(B)		Web Programming									
CA-105(A)	Skill Based	Java Programming (Core Java)	04	-	04	40	-	60	-	100	04
CA-105(B)		Object Oriented Programming using C++									
CA LAB-I	Core	LAB on DBMS	-	02	02	-	20	-	30	50	02
CA LAB-II	Core	LAB on OS (Linux)	-	02	02	-	20	-	30	50	02
CA LAB-III(A)	Core	LAB on COPS		02	02	-	20	-	30	50	02
CA LAB-III(B)		LAB on Web Programming									
CA LAB-IV(A)	Skill Based	LAB on Java Programming		02	02	-	20	-	30	50	02
CA LAB-IV(B)		LAB on C++ Programming									
AC-101	Audit Course	Practicing Cleanliness	-	02	02	-	100	-	-	100	02

Semester II

Total Credit for Semester II: 30 (T = Theory: 16; P = Practical: 6; Skill Based: 6; Audit Course:2)

	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination				Total	Credits
						Internal		External			
			T	P	Total	T	P	T	P		
CA-201	Core	Advanced Software Development Methodologies	04	-	04	40	-	60	-	100	04
CA-202	Core	Mathematical Foundations of Computer: Science	04	-	04	40	-	60	-	100	04
CA-203	Core	Data Structures and Algorithms	04	-	04	40	-	60	-	100	04
CA-204(A)	Core	Machine Learning	04	-	04	40	-	60	-	100	04
CA-204(B)		Digital Image Processing & Computer Vision									
CA-205(A)	Skill Based	Advanced Java (Technologies)	04	-	04	40	-	60	-	100	04
CA-205(B)		Python Programming									
CA LAB-V	Core	LAB on Advanced Software Development Methodologies	-	02	02	-	20	-	30	50	02
CA LAB-VI	Core	LAB on Data Structures and Algorithms	-	02	02	-	20	-	30	50	02
CA LAB-VII(A)	Core	LAB on Machine Learning		02	02	-	20	-	30	50	02
CA LAB-VII(B)		LAB on Digital Image Processing & Computer Vision									
CA LAB-VIII(A)	Skill Based	LAB on Advanced Java (Technologies)		02	02	-	20	-	30	50	02
CA LAB-VIII(B)		LAB on Python programming									
AC-201	Audit Course	Personality and Cultural Development	-	02	02	-	100	-	-	100	02

Semester-wise Course Structure of M.C.A. II (w.e.f. A Y 2020-2021)

Semester III

Total Credit for Semester I: 30 (T = Theory: 16; P = Practical: 6; Skill Based: 6; Audit Course:2)

	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination					Credits
						Internal		External		Total	
			T	P	Total	T	P	T	P		
CA-301	Core	Compiler Construction	04	-	04	40	-	60	-	100	04
CA-302	Core	Design and Analysis of Algorithms	04	-	04	40	-	60	-	100	04
CA-303	Core	High Performance Computing Paradigms and Applications	04	-	04	40	-	60	-	100	04
CA-304(A)	Core	Natural Language Processing	04	-	04	40	-	60	-	100	04
CA-304(B)		AI in Practice with Python									
CA-304(C)		Data Analytics	04	-	04	40	-	60	-	100	04
CA-305(A)	Skill Based	Mobile Application Development (Android Programming)	04	-	04	40	-	60	-	100	04
CA-305(B)		Microsoft .Net Technologies									
CA-305(C)		Ruby on Rails									
CA LAB-IX	Core	LAB on Design and Analysis of Algorithms	-	02	02	-	20	-	30	50	02
CA LAB-X	Core	Lab on High Performance Computing Paradigms and Applications	-	02	02	-	20	-	30	50	02
CA LAB-XI(A)	Core	Lab on Natural Language Processing		02	02	-	20	-	30	50	02
CA LAB-XI(B)		Lab on AI Practice using Python									
CA LAB-XI(C)		Lab on Data Analytics									
CA LAB-XII(A)	Skill Based	Lab on Android Programming		02	02	-	20	-	30	50	02
CA LAB-XII(B)		Lab on Microsoft .Net Technologies									
CA LAB-XII(C)		Lab on Ruby on Rails									
AC-301	Audit Course	Technology + Value Added Course	-	02	02	-	100	-	-	100	02

Semester IV

Total Credit for Semester IV: 12

	Course Type	Title of the Course	Contact Hours/Week	Distribution of Marks for Examination			Credits
				Internal	External	Total	
CA-401	Project	Full time Industrial Training	Students contact to teachers through E-mail, AView Software and other ICT technologies throughout the Semester	-	300	300	12

Program at a Glance

Name of the program (Degree)	:	Master in Computer Applications (MCA)
Faculty	:	Science and Technology
Duration of the Program	:	Two years (four semesters)
Medium of Instruction and Examination	:	English
Examination Pattern	:	60% (External Assessment) + 40% (Internal Assessment)
Passing Standard	:	Separate Passing for internal as well as external assessment (40%).
Evaluation mode	:	CGPA
Total Credits of the program	:	102 (102 core credits including 12 credits of project/dissertation, 18 skill enhancement credits, 24 subject elective credits and 06 audit credits)

Programme Specific Objectives:

- Prepare students to become computer professionals with comprehensive knowledge and skills to produce software for emerging requirement
- Prepare students to become continuous learner with aptitude for teaching and research with societal focus
- Prepare students who will achieve peer-recognition; as an individual or in a team; through demonstration of good analytical, design and implementation skills.

Semester-I

Course Code: **CA-101**	**Database Management System (DBMS)**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
1) Introduction to the basic concepts of database management systems, learning to design databases using ER modelling, and decomposing data based on functional dependencies
2) Understand Relational databases, SQL, Transaction management, Query processing, concurrency control and recovery system.
3) Describe and discuss selected advanced database topics such as distributed database and XML and Web data.

Unit-I	[05]	Max Marks:08

Introduction: 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.

Unit-II	[10]	Max Marks:12

Database Design and E-R Model: 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).

Unit-III	[12]	Max Marks:20

Relational Databases: 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.

Unit-IV	[10]	Max Marks:22

Transaction Management and Query Processing: 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.

Unit-V	[10]	Max Marks:16

Concurrency Control and Recovery System: 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

Unit-VI	[13]	Max Marks:22

Advanced Topics in Databases: 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.

Distributed Databases: Overview, Homogeneous and Heterogeneous Databases, Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control, Cloud based Databases.

References:
[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.
[2] C. J .Date, A. Kannan and S. Swamynathan (2006), "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006, ISBN:978-81-7758-556-8
[3] Silberschatz, H.F.Korth, and S.Sudarshan (2011), "Database System Concepts", TMH Publications, Sixth Edition, 2011, ISBN: 978-007-132522-6.
[4] Ramez Elmasri, Shamkant B. Navathe (2011), "Fundamentals of Database Systems" Seventh Edition, Pearson Education, 2011, ISBN: 978-0-13-397077-7.

Course Outcome:
After completion of this course students shall be able to-
1. Apply the relational model, specify integrity constraints, and explain how to create a relational database using an ER diagram and normalization techniques.
2. Apply SQL to create, query and manipulate relational databases.
3. Determine partitioning and distribution of data across networked nodes of a DBMS and data optimization in a distributed environment.

Course Code: **CA-102**	**Operating Systems**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
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.
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
3) To study the need for special purpose operating systems with the advent of new emerging technologies

Unit-I	**[04]**	Max Marks:0**8**
Introduction: review of computer organization, introduction to popular operating systems like UNIX, Windows, etc., OS structure, system calls, functions of OS, evolution of Oss.		
Unit-II	**[03]**	Max Marks:0**6**
Computer organization interface: using interrupt handler to pass control between a running program and OS.		
Unit-III	**[08]**	Max Marks:**12**
Concept of a process: states, operations with examples from UNIX (fork, exec), Process scheduling, interprocess communication (shared memory and message passing), UNIX signals.		
Unit-IV	**[04]**	Max Marks:0**6**
Threads: multithreaded model, scheduler activations, examples of threaded programs.		
Unit-V	**[06]**	Max Marks:**10**
Scheduling: multi-programming and time sharing, scheduling algorithms, multiprocessor scheduling, thread scheduling (examples using POSIX threads).		
Unit-VI	**[08]**	Max Marks:**12**

Process synchronization: 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.).		
Unit-VII	**[06]**	Max Marks:**10**
Deadlocks: modelling, characterization, prevention and avoidance, detection and recovery.		
Unit-VIII	**[07]**	Max Marks:**12**
Memory management: 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.		
Unit-IX	**[07]**	Max Marks:**12**
Secondary storage and Input/Output: 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		
Unit-X	**[04]**	Max Marks:**06**
Protection and security: Illustrations of security model of UNIX and other Oss. Examples of attacks.		
Unit-XI	**[03]**	Max Marks:**06**
Epilogue: Pointers to advanced topics (distributed OS, multimedia OS, embedded OS, real-time OS, OS for multiprocessor machines).		
All above topics shall be illustrated using UNIX as case-studies.		

References:
1. 1 Abraham Silberschatz, Peter B. Galvin, Greg Gagne (2009), Operating System Concepts, 8th Ed., John Wiley ISBN 0-471-69466-5.
2. William Stallings (2014), Operating Systems: Internals and Design Principles. Pearson, 8th Ed. ISBN-13: 978-0-13-230998-1
3. AS Tanenbaum (2009), Modern Operating Systems, 3rd Ed., Pearson. ISBN: 0135013011
4. AS Tanenbaum, AS Woodhull (2006), Operating Systems Design and Implementation, 3rd Ed., Prentice Hall ISBN-10: 0131429388
5. M. J. Bach (1986), Design of the Unix Operating System, Prentice Hall of India ISBN0. -13-201757-1 025

Course Outcome:
After completion of this course, students shall be able to:
1) Analyse design aspects and data structures/policies/algorithms used for file subsystem, memory subsystem, process subsystem and i/o subsystem of Unix OS.
2) Differentiate between threads and processes and compare different processor scheduling algorithms
3) Identify the need to create the advance and special purpose operating system.

Course Code: **CA-103**	**Fundamentals of Artificial Intelligence**	*Clock Hours:* **60** *Total Marks:* **100**

*Course Objectives***:**
1) Gain a historical perspective of AI and its foundations.
2) Study the concepts of Artificial Intelligence and investigate applications of AI techniques in intelligent agents

3) Learn various peculiar search strategies used in AI and use of them in solving problems using Artificial Intelligence.		
Unit-I	**[08]**	Max Marks:**10**
What is AI?: Overview and Historical Perspective, Turing test, Physical Symbol Systems and the scope of Symbolic AI, AI Agents.		
Unit-II	**[06]**	Max Marks:**10**
Uninformed Search: State Space Representation, Depth First Search, Breadth First Search, DFID.		
Unit-III	**[08]**	Max Marks:**12**
Informed Search: Best First Search, Hill Climbing, Beam Search, Tabu Search.		
Unit-IV	**[08]**	Max Marks:**15**
Randomized Search: Simulated Annealing, Genetic Algorithms, Ant Colony Optimization.		
Unit-V	**[08]**	Max Marks:**12**
Problem Decomposition: Goal Trees, AO*, Rule Based Systems, Rete Net.		
Unit-VI	**[06]**	Max Marks:**12**
Game Playing: Minimax Algorithm, AlphaBeta Algorithm, SSS*.		
Unit-VII	**[08]**	Max Marks:**14**
Mathematical Logic and Inferences: Propositional Logic, First Order Logic, Soundness and Completeness, Forward and Backward chaining.		
Unit-VIII	**[08]**	Max Marks:**15**
Planning: Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graphplan,		
References: 1. Deepak Khemani (2013). A First Course in Artificial Intelligence, McGraw Hill Education (India). 2. Elaine Rich and Kevin Knight (1991). Artificial Intelligence, Tata McGraw Hill. 3. Stuart Russell and Peter Norvig (2009). Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall.		

Course Outcome:
After completion of this course students shall be able to-
    1) Identify problems that are amenable to solution by AI methods.
    2) Identify appropriate AI methods to solve a given problem.
    3) Design smart system using different informed search / uninformed search or heuristic approaches.		

Course Code: **CA-104(A)**	**Computer Programming and Problem Solving**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
    1) To introduce the foundations of computing, programming and problem-solving.
    2) To develop logical ability for problem-solving.
    3) To develop basic programming skills necessary for coding.		

Unit-I	**[10]**	Max Marks:**16**
Introduction to problem solving 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.		
Unit-II	**[10]**	Max Marks:**18**
Solving Problems with iterations verses Recursion		

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.		
Unit-III	**[12]**	Max Marks:**20**
Solving Problems with Vector and Matrices		
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.		
Unit-IV	**[12]**	Max Marks:**18**
Solving Problems with Dynamic Memory		
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		
Unit-V	**[08]**	Max Marks:**14**
Structural approach verses object oriented approach		
Introduction to object oriented paradigms, Components of Structural approach and object oriented approach, Structural approach verses object oriented approach		
Unit-VI	**[08]**	Max Marks:**14**
Solving Problems with code reuse		
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.		

Course Code: **CA-104(B)**	**Web Programming**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
     1) To understand web designing using HTML/CSS.
     2) To use JavaScript for scripting.
     3) To understand staic/dynamic API using JSON/JQuery/AngularJS.		

Unit-I	**[10]**	Max Marks:**16**
HTML: Introduction to HTML, Doctype, Namespace, Meta Tag, Script Tag, Images, Tables, Div, Paragraph, Span, Pre Tags, Anchor Links and Named Anchors, Line Breaks and Horizontal Lines. Lists, Object Tag, Iframe Tag, Form Tag, Form Tag, POST and GET Method, Fieldset and Legend,		

Text input, Text area, Checkbox and Radio Button, Dropdown, List, File Upload and Hidden Fields, Submit, Image, Normal, Reset Button, Creating a Live Website with Form

CSS: Introduction to CSS 2.1, CSS Selectors: Universal Selector, ID Selector, Tag Selector, Class Selector, Sub Selector, CSS Properties: Type Properties, Background Properties, Block Properties, Box Properties, List Properties, Border Properties, Positioning Properties, Conversion of Table to CSS Layout, CSS Menu Design (Horizontal, Vertical), External and Inline CSS, Introduction to CSS 3, New CSS 3 Selectors, New CSS3 Properties, CSS Gradients, Opacity Property, ,Transition effect, Transform effect, Animation effects, Css Media Queries, Creating a Live Website with CSS

HTML 5: Introduction to HTML5, XHTML vs HTML5, Features of HTML5, HTML5 DocType, New Structure Tags, Section, Nav , Article, Aside, Header, Footer, New Media Tags, Canvas and Svg Tag, Introduction to HTML5 Forms, voice search

JavaScript: 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

Unit-II	[10]	Max Marks:**18**

 Bootstrap :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

Unit-III	[12]	Max Marks:**20**

AngularJS: Introduction to AngularJS, Structuring AngularJS application, MVC in AngularJS, AngularJS routing, AngularJS services

Unit-IV	[12]	Max Marks:**18**

JQuery: Basics of jQuery, jquery selection and events, jQuery Effects, jquery traversal and manipulation, Data attributes and templates, jQuery Plugins, JQuery / Google Web Toolkit

Unit-V	[08]	Max Marks:**14**

Node.js: 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.

Unit-VI	[08]	Max Marks:**14**

 JSON: 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

References:
1. HTML, CSS, & JavaScript for Dummies. Tittel, E., Holland, E., Minnick, C. (2018). John Wiley & Sons.
2. HTML5 Programmer's Reference. Reid, J. (2015). Apress.
3. Bootstrap: Responsive Web Development. Spurlock, J. (2013). OReilly Media.
4. Professional AngularJS. Karpov, V., Netto, D. (2015). Wiley.
5. Web Development with JQuery. York, R. (2015). Wiley.
6. Professional Node.js: Building Javascript Based Scalable Software. Teixeira, P. (2012). Wiley.
7. Beginning JSON. Smith, B. (2015). Apress.

Course Code: **CA-105(A)**	**Java Programming (Core Java)**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
1. Understand Fundamental concepts of object oriented programming using Java technology.
2. Java applications development using polymorphism, inheritance, interfaces and inner classes and multi-threading.
3. GUI applications and event driven applications development.

Unit-I	**[08]**	Max Marks:**5**

An Introduction to Java: 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

Unit-II	**[08]**	Max Marks:**10**

Introduction to OOPs: 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

Unit-III	**[10]**	Max Marks:**20**

Inheritance, Polymorphism and interfaces: 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

Unit-IV	**[10]**	Max Marks:**20**

Multithreading and Exceptions: 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

Unit-V	**[14]**	Max Marks:**25**

Graphics Programming and event handling : 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 (Flow Layout, Border Layout, Card Layout, Grid Layout, GridBag Layout), Basics of Event Handling, Listeners and Listener Methods, Mouse Events, Keyboard Events, AWT Event Hierarchy

Unit-VI	**[10]**	Max Marks:**20**

Streams, Files and JDBC: 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.

Course Code: **CA-105(B)**	**Object Oriented Programming using C++**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:

The objectives of the course are:
1) To familiarize the Object-Oriented Programming (OOP) concepts, such as abstraction, encapsulation, instances, initializations, polymorphism, overloading, inheritance etc.
2) To write programs to solve problems using generic programming constructs such as templates and using standard template library.
3) To understand and know the importance of pointers and learn file handling and exception handling in real-world problems.

Unit-I	**[15]**	Max Marks:**15**

Fundamentals:

Object-Oriented Programming (OOP): Need, Object Oriented Programming Paradigm, Benefits of OOP, C++ as object-oriented programming language.

C++ programming Basics: Data types, Enumerations, Arrays, Strings, Pointers and references, Control structures.

Functions: 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.

OOP Concepts: 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.

Unit-II	**[10]**	Max Marks:**10**

Inheritance:

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.

Unit-III	**[10]**	Max Marks:**20**

Polymorphism:

Operator Overloading: concept of overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Keywords 'explicit' and

'mutable'. *Pointers-* indirect ion Operators, Memory Management: new and delete, Pointers to Objects. *Virtual Functions*: concept, pure virtual functions and abstract classes, arrays in polymorphism, late binding, Function pointers, Debugging Pointers, Dynamic Pointers, smart pointers.

Unit-IV	[10]	Max Marks:**20**

Files and Streams:

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.

Unit-V	[10]	Max Marks:**20**

Templates and Exception Handling:

Templates: Function templates, Template specialization, Class templates, Non-type parameters for templates, template, and inheritance, The typename and export keywords. *Exception Handling*: Other error handling techniques, Exceptions, Exception handling in C++, rethrowing an exception, exception specifications, processing unexpected exceptions, stack unwinding, exception handling in constructors, destructors.

Unit-VI	[05]	Max Marks:**15**

Standard Template Library (STL):

Introduction to STL: Containers, algorithms, adaptors, and iterators, *Containers*: Sequence container and associative containers, *Adaptors*: container adapters, iterator adaptors, *Algorithms*: basic searching and sorting algorithms, min-max algorithm, set operations, *Iterators*: input, output, forward, bidirectional and random access.

References:

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, RajivShah, Vijay Mukhi's The 'C' Odyssey C++ and Graphics-The future of C, BPB publications, First Edition

Course Outcome:

After completion of this course students shall be able to-
1. Understand and use the basic programming constructs of C++ and manipulate various C++ datatypes, such as arrays, strings, and pointers.
2. Manage memory appropriately using proper allocation/deallocation procedures.
3. Write small-scale C++ programs using the above skills.

Course Code: **CA LAB-I**	**LAB on DBMS**	*Total Marks:* **50**

Course Objectives:
   1) Provides foundation knowledge in database concepts, technology and practice to prepare students into expert database application developers.
   2) Strong practice in SQL programming through a variety of database problems.
   3) Develop database applications using front-end tools and back-end DBMS.

1. Creating database tables and using data types.
      • Create table , Modify table, Drop table

2. Practical Based on Data Manipulation.
   • Adding/Modify/Delete data using Insert/ Update/ Delete
3. Practical Based on Implementing the Constraints.
   • NULL and NOT NULL, Primary Key Constraint, Foreign Key Constraint
   • Unique Constraint, Check Constraint, Default Constraint
4. Practical for Retrieving Data Using following clauses.
   • Simple select clause
   • Accessing specific data with Where Clause
   • Ordered By/ Distinct/Group By Clause
5. Practical Based on Aggregate Functions.
   • AVG, COUNT, MAX, MIN, SUM, CUBE
6. Practical Based on implementing all String functions.
7. Practical Based on implementing Date and Time Functions.
8. Practical Based on implementing use of UNION, INTERSECTION, SET DIFFERENCE.
9. Implement Nested Queries & all types of JOIN operation.
10. Practical Based on performing different operations on a view.
11. Practical Based on implementing use of Procedures.
12. Practical Based on implementing use of Triggers
13. Practical Based on implementing Cursor.
14. Demonstrate Database connectivity with front end tools like – VB.NET, C#.NET, JAVA etc.
15. Practical based on creating Data Reports.
16 Design entity relationship models for a business problem and develop a normalized database
structure

Course Outcome:
After completion of this course students shall be able to-
   1) Design and implement a database schema for a given problem-domain
   2) Create and maintain tables using PL/SQL, Populate and query a database using SQL
      DML/DDL commands and programming PL/SQL including stored procedures, stored
      functions, cursors, triggers.
   3) Application development using PL/SQL & front-end tools.

Course Code: **CA LAB-II**	**LAB on OS (Linux)**	*Total Marks:* **100**

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

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)

3. 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

4. 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.

Course Code: **CA LAB-III**(A)	**LAB on Computer Programming and Problem Solving(COPS)**	*Total Marks:* **50**
Course Objectives: 1) To represent the problems using mechanism like charts. 2) To express logic as an algorithm. 3) To convert logic in programs.		
Laboratory Requirements: **OS:** Windows/Linux, **Programming Language:** Any programming language **Instructions** • All practical assignments must be designed on paper, logic should be demonstrated and pseudocode is to be written. • No barrier of programming language for code conversion of the assignments. • Problems and puzzles in theory are the assignments for the followings.		

1.	Simple exercises and examples to introduce to the computing environment and usage.
2.	Simple exercises and examples of functional programming
3.	Puzzle solving using iterations
4.	Problem solving using recursion
5.	Programming for vectors and multidimensional data
6.	Dynamic memory and problem solving.
7.	Assignments on Solving Problems with code reuse

Course Outcome:

After completion of this course students shall be able to-
1) Construct logic for the problems.
2) Write algorithms and able to draw logic on paper.
3) Write code for the logic developed.

Course Code: **CA LAB-III(B)**	**LAB on Web Designing**	*Total Marks:* **50**

Course Objectives:
1) To work in web designing using HTML/CSS.
2) To use Bootstrap for designing.
3) To design dynamically using JSON/JQuery/AngularJS.

1. Design a website with HTML Form.

2. Design a website using CSS 2.1 and CSS3.

3. Design a website with HTML5.

4. Design a dynamic web form with validations using JavaScript.

5. Design a website with Bootstrap.

6. Design a dynamic website with AngularJS.

7. Demonstrate the use of jQuery in a website.

8. Demonstrate the use of Node.js in a website.

9. Demonstrate the use of JSON in a website.

10. Design a dynamic website using demonstrating the web technologies (HTML, JavaScript, Bootstrap, Angular JS, JQuery).

Course Outcome:

After completion of this course students shall be able to-
1) Develop Web site/App.
2) Use Bootstrap/Javascript to make design and scripting.
3) Make Web site dynamic using AngularJS/JSON/JQurey.

Course Code: **CA LAB-IV(A)**	**: LAB on Java Programming**	*Total Marks:* **50**

Course Objectives:
1) Programming using inner classes and inheritance, polymorphism and interfaces
2) Use various swing components and handle several events in the development of GUI applications
3) Use JDBC and package creation

1] Write a program that demonstrate program structure of java with use of arithmetical and logical implementation.

2] Write a program that demonstrate string operations using String and StringBuffer class.

3] Write a program that demonstrate inner class and static fields.

4] Write a program that demonstrate inheritance, polymorphism.

5] Write a program that demonstrate 2D shapes on frames.

6] Write a program that demonstrate color and fonts.

7] Write a program to illustrate use of various swing components.

8] Write a program that demonstrate use of dialog box and menus.

9] Write a program that demonstrate event handling for various types of events.

10] Write a program to illustrate multithreading.

11] Write a program to illustrate exception handling.

12] Write a program to demonstrate use of File class.

13] Write a program that demonstrate JDBC on application.

14] Write a program that demonstrate package creation and use in program.

Course Outcome:

After completion of this course students shall be able to-

1) Write java programms using inner classes and static fields in implementation of Java application
2) Develop Java application for GUI development and event handling.
3) Develop database application using JDBC.

Course Code: **CA LAB-IV(B)**	**LAB on C++ Programming**	*Total Marks:* **50**

Course Objectives:

1) Apply object-oriented approaches to software problems in C++
2) Apply exception handling techniques to software problems in C++
3) Apply generic programming approaches using templates and efficiently use standard template library in software development

1. Write program to demonstrate class, use of constructor, constructor overloading and destructor.
2. Write program to demonstrate use of arrays, strings, pointers, constants, and references.
3. Write program to demonstrate use of operator overloading.
4. Write program(s) to demonstrate use of inheritance.
5. Write program to demonstrate use of compile time and runtime polymorphism.
6. Write program to demonstrate use of friend function and friend class.
7. Write program to demonstrate use of virtual class.
8. Write program to demonstrate use of static data member and static member function.
9. Write program to demonstrate file handling.
10. Write program to demonstrate use of function templates.
11. Write program to demonstrate use of class templates.
12. Write program to demonstrate use of exception handling.
13. Write program to demonstrate command line arguments.
14. Write program(s) to demonstrate use of STL.

Course Outcome:

After completion of this course students shall be able to-

1) Develop logic of a program for solving real time problems and isolate and fix common errors in C++ programs

2) Understand the object-oriented approach for the program development and make use of the OOP concepts (data abstraction, encapsulation, polymorphism, overloading, and inheritance) of C++ appropriately in problem solving.
3) Create applications using the STL library.

Semester-II

Course Code: **CA-201**	**Advanced Software Development Methodologies**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:		

Course Objectives:
The objectives of the course are:
1) To introduce git for software development
2) To learn the principles and practices associated with each of the agile development methods.
3) To apply the principles and practices of agile software development on a project of interest and relevance to the student.

Unit-I	**[10]**	Max Marks:**16**

Git & Version Control: Basic Concepts, Environment setup, Life Cycle, Branches & Merging, working with local repository and Remote Repository.

Unit-II	**[10]**	Max Marks:**18**

Agile Methodology: 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.

Unit-III	**[12]**	Max Marks:**20**

Agile Processes: Lean production - SCRUM, Crystal, Feature Driven Development, Adaptive Software Development, and Extreme Programming: Method overview, lifecycle, work products, roles and practices.

Unit-IV	**[12]**	Max Marks:**18**

Agility And Knowledge Management: 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).

Unit-V	**[08]**	Max Marks:**14**

Agility and Requirements Engineering: 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.

Unit-VI	**[08]**	Max Marks:**14**

 Agility and Quality Assurance: 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*

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.

Course Code: **CA-202**	**Mathematical Foundations of Computer Science**	Clock Hours: **60** Total Marks: **100**

Course Objectives:

The objectives of the course are:

1. To build the foundation of computer algorithms using mathematical base.
2. To apply statistical measures on the data and represent it graphically.
3. To relate practical examples to the probability theory and probability distributions to build the foundation for machine learning.

Unit-I	**[10]**	Max Marks:**20**

Induction and Recursion: Mathematical Induction, Strong Induction and Well Ordering, Recursive Algorithms, Program Correctness, The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations.

Unit-II	**[15]**	Max Marks:**20**

Advance Counting Techniques:

Recursive Relations, The Towers of Hanoi, Merge Sort, Linear Recurrences, Solving Linear Recurrence Relations, Divide-and-Conquer Recurrences, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

Unit-III	**[10]**	Max Marks:**15**

Statistics: Population, sample, parameters, and statistics: definition, types, applications, Data Presentation: Classification of data, Frequency distribution, Cumulative and Relative frequency distribution, Descriptive Statistics: mean, median, mode, range, quartile deviation, standard deviation, variance, Graphical statistics

Unit-IV	**[10]**	Max Marks:**15**

Probability: Making decisions under uncertainty, Classical definition of Probability, Events and their Outcomes, Rules of Probability, Probability axioms, Joint and Conditional probability, independence, and Bayes theorem,

Unit-V	**[10]**	Max Marks:**20**

Probability Distributions: Random variables (discrete and continuous), Probability mass function, Distributions: Binomial, Poisson, Probability density function, Distributions: Uniform, Exponential, Normal.

Unit-VI	**[05]**	Max Marks:**10**

Stochastic Processes: Definitions and classifications of Stochastic Processes, discrete and continuous Markov models, Chapman-Kolmogorov equation.

References:

1. Kenneth H. Rosen, Discrete Mathematics, and its Applications 6th Ed, Tata McGraw-Hill Publishing Company Limited, New Delhi, 2007 ISBN 10: 0070681880
2. Michael Baron (2014) Probability and Statistics for Computer Scientists Second Edition, CRC press. ISBN: 978-1-4822-1410-9
3. Goon A.M., Gupta M.K., Dasgupta. B. (2001), Fundamentals of Statistics, Volume I and II, World Press, Calcutta.
4. Ross, S. (2005). Introduction to Probability Models, (6th Ed. Academic Press). ISBN 978-0-12-375686-2

5. Medhi, J. (1994). Stochastic Processes, (2nd Ed. New Age Publisher) ISBN : 978-93-86286-48-2

Course Outcome:

After completion of this course students shall be able to-

1. Identify, formulate, and develop solutions to computational challenges.
2. Analyze the behavior of the data, model the data using statistical measures and represent it graphically on paper without using available computerized tools.
3. Apply mathematical foundations, probability theory in the modeling and design of computational systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

Course Code: **CA-203**	**Data Structures and Algorithms**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
1) To impart the basic concepts of data structures and algorithms
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
3) To understand concepts about searching, sorting and hashing techniques

Unit-I	**[08]**	Max Marks:**06**

Introduction to Data Structures and Algorithms: 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

Unit-II	**[08]**	Max Marks:**15**

Array: Storage representation, operations and applications (Polynomial addition and subtraction) **Stack**: operations and applications (infix, postfix and prefix expression handling), **Queue**: operations and applications, Circular Queues: operations and applications, Concept of Double ended Queue and Priority Queue, Linked representation of stack and queue.

Unit-III	**[10]**	Max Marks:**12**

Linked Lists: Operations and Applications of Linear linked list (Polynomial addition and subtraction), Circular linked list and Doubly linked list.

Unit-IV	**[11]**	Max Marks:**21**

Trees: 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

Unit-V	**[11]**	Max Marks:**18**

Graphs and Their Applications: 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.

Unit-VI	**[12]**	Max Marks:**18**

Hash Table: Hash Function, Collision and its Resolution, Separate Chaining, Open Addressing (linear probing, quadratic probing, double hashing), Rehashing, Extendible Hashing **Searching:** Linear Search and Binary Search (array/binary tree). **Sorting**: 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++, 2nd Edition, Pearson Education ISBN: 978-9332549319, 9332549311
3. Balagurusamy, E., (2013), Data Structures using C, 1st Edition, Mcgraw Hill Education, ISBN: 978-1259029547, 1259029549
4. Weiss, M.A., (2002), Data Structures and Algorithm Analysis in C, 2nd 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, 2nd 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.

Course Code: **CA-204(A)**	**Machine Learning**	*Clock Hours:* **60** *Total Marks:* **100**

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.

Unit-I	**[08]**	**Max Marks:10**
Introduction: Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation		
Unit-II	**[08]**	Max Marks:**15**
Regression(Linear, Lasso, Ridge), Decision trees, overfitting		
Unit-III	**[09]**	Max Marks: **15**
Instance based learning, Feature reduction, Collaborative filtering-based recommendation		
Unit-IV	**[08]**	Max Marks: **15**
Probability and Bayes learning		
Unit-V	**[09]**	Max Marks:**15**
Logistic Regression, Support Vector Machine, Kernel function and Kernel SVM		
Unit-VI	**[09]**	Max Marks: **15**

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

Unit-VII	**[09]**	Max Marks: **15**

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.

Course Code: **CA-204(B)**	**Digital Image Processing & Computer Vision**	*Clock Hours:* **60** *Total Marks:* **100**

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.

Unit-I Introduction to Digital Image Processing & Applications	**[08]**	**Max Marks:12**

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.

Unit-II Image Enhancement	**[10]**	**Max Marks:18**

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,

Unit-III Color Image Processing	**[4]**	**Max Marks:10**

Color Fundamentals, Color Models, Pseudocolor Image Processing, Basics of Full-Color Image Processing, Color Transformations, Smoothing and Sharpening.

Unit-IV Introduction to computer vision	**[10]**	**Max Marks: 15**

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.

Unit-V Basics of video processing	**[10]**	**Max Marks:15**

Stereo – Epi-polar geometry, correspondence, triangulation, Disparity maps, Background subtraction techniques – frame differencing, Gaussian Mixture Modelling (GMM), Object localization and processing: Contours, edges, lines, skeletons.

Unit-VI Object Recognition	[8]	Max Marks: 12

Object Modeling, Bayesian Classification, Feature Selection and Boosting, Scene and Object Discrimination.

Unit-VII Motion and Tracking	[10]	Max Marks: 18

Motion detection and tracking of point features, optical flow, SURF, SIFT. Tracking- Kalman filter, Particle Filter, Comparison of deterministic and probabilistic methods condensation, tracking humans, multi-frame reconstruction under affine and perspective projection geometry.

References:
1. R.C.Gonzalez & R.E.Woods, Digital Image Processing, Pearson Education, 3rd edition, ISBN. 13:978-0131687288
2. S. Jayaraman Digital Image Processing TMH (McGraw Hill) publication, ISBN-13:978-0-07-0144798
3. Gonzalez, Woods & Steven, Digital Image Processing using MATLAB, Pearson Education, ISBN-13:978-0130085191
4. .Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
5. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.

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.

Course Code: **CA-205(A)**	**Advanced Java (Technologies)**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
The objectives of the course are:
    1) To become familiar with the advanced features of Java Language as generic programming, collection framework.
    2) To understand RMI technology and concept of reusable components using JavaBeans and EJB.
    3) To understand Java Servlets, Java server Pages (JSP) technology and Strut & Hibernate technology

Unit-I	[06]	Max Marks:10

Generic programming: Generic Class, Generic Method, Generic Interface; **The Collection framework:** Collection Object, Retrieving elements from Collection, HashSet Class, LinkedHashSet Class, Stack Class, LinkedList Class, ArrayList Class, Vector Class, HashMap

Class , Arrays Class; Object Wrappers and Autoboxing; **Distributed Computing:** Remote Method Invocation- Introduction, Architecture, RMI Object services, stub and Skeleton, Steps of developing an RMI system		
Unit-II	**[10]**	Max Marks:**10**
Java Bean: Concepts, Writing process, Applications, Properties and Events, Property Editors, Customizer , Persistence; **Enterprise JavaBeans:** Introduction, Specification, Architecture, Container, Types, Life cycle, Applications		
Unit-III	**[12]**	Max Marks:**20**
Servlets: Concepts, Architecture, Servlet Container Writing Process, API, Life Cycle, Hierarchy, ServletConfig, ServletContext, Programming and deployment, Servlet and HTML Form, Session Management, JDBC ;		
Unit-IV	**[10]**	Max Marks:**20**
Java Server Pages: Introduction, JSP Containers, Architecture, JSP and Servlets. Life Cycle of JSP Page, JSP Processing, JSP Application Design with MVC, 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.		
Unit-V	**[10]**	Max Marks:**20**
MVC: Introduction different types of logic in Java based Web Application, Advantages and Disadvantages of MVC Architecture, MVC pattern Layer: Model, View and Controller. **Strut:** Introduction, Understanding Scopes, Custom Tags, The MVC Design Pattern, Simple Validation, Processing Business Logic, Basic Struts Tags, Configuring Struts, validation framework.		
Unit-VI	**[12]**	Max Marks:**20**
Overview of Hibernate, Hibernate Architecture, Hibernate Mapping Types, Hibernate O/R Mapping, Hibernate Annotation, Hibernate Query Language.		
References: 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. SoumadipGhosh, "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 - 1. Develop applications using advanced features of Java Language, RMI, JavaBeans and EJB. 2. Use Java Servlets, JSP to develop web applications 3. Use Strut and Hibernate in development of dynamic web applications**.**		

Course Code: **CA-205(B)**	**Python Programming**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
1) Introduce to python programming: data types, operators, conditional and logical statements, control structures, writing user defined functions and file handling. To introduce to OOP through python, regular expressions, exception handling and GUI constructs for web.
3) To study advance topics in python viz., lambda functions, functional programming tools, using and configuring modules etc.

Unit-I	**[12]**	Max Marks:15

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 wile loop

Unit-II	**[08]**	Max Marks:**15**

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.

Unit-III	**[12]**	Max Marks:**20**

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 Constructers, Python built-in class functions, Inheritance, Overloading Methods, Overriding methods, Data Hiding, Search Algorithms, Sorting Algorithms, Hash Tables

Unit-IV	**[08]**	Max Marks:**15**

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.

Unit-V	**[12]**	Max Marks:**20**

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.

| **Unit-VI** | **[08]** Max Marks: 15 |

Python MySQL: Environment setup, Database Connection, Creating New Database, Creating Tables, Insert Operation, Read Operation, Update Operation, Join Operation, Performing Transactions, Simple Flask Operation, Simple Django Framework.

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 Pyhon", 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, 3rd 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.

Course Code: **CA LAB-V**	**CA LAB-V: LAB on Advanced Software Development Methodologies**	*Total Marks:* **50**

Course Objectives:
1) To setup GitHub and use git.
2) To understand agile process.
3) To use agile methodology.

Practical's on Git
1. Set up a GitHub account and send an email message to friend telling to your friend about the account.
2. Clone the repository at https://github.com/KBCNMU.
3. Use git add to add that file to the repository. Use git commit to commit your change.
4. Use git push to send your change to the primary repository. Look online to see if your change has been pushed.
5. Use git pull to get your partners change into your repository. Verify that you have the new file.

6. Use git log or git log | less to see a list of changes to the repository.
7. Use git branch to create a new branch and git checkout to switch to the branch.
8. Use git merge to merge you changes to the primary branch.
9. Demonstrate the use of git diff command

Practical's on Agile

Develop a mini project using any technology. Document the process Agile methodology.

Course Outcome:

After completion of this course students shall be able to-
   1) Use GitHub and make repository using Git.
   2) Apply agile software development process.
   3) Develop a project using agile methodology.

Course Code: **CA Lab-VI**	**LAB on Data Structures and Algorithms**	*Total Marks:* **50**

Course Objectives:
   1) Solve real-world problems by reasoning about data structure choices, choose appropriate implementations.
   2) To make the students write various programs and ADTS for all data structures.
   3) Students will learn to write, debug, and test large programs systematically.

Implementation of programs based on the following
   • Arrays
   • Multidimensional Arrays, Matrices
   • Stacks, Polish Notation
   • Queues
   • Deques
   • Linear Linked List, Circular Linked List, Doubly Linked List
   • Polynomial Addition/Subtraction

Implementation of programs based on Trees
   • Binary Search Tree
   • In-order, Pre-order and Post-order Traversals
   • Heap Tree

Implementation of programs based on Graphs
   • Depth First Traversal
   • Breadth First Traversal
   • Obtaining Shortest Path (Dijkstra and Floyd-Warshall)
   • Minimum spanning tree (Kruskal and Prim)

Implementation of programs for Hash Table, Searching and Sorting techniques
   • Hash Table
   • Linear and Binary Search (using array)
   • Bubble sort
   • Selection sort
   • Insertion sort
   • Radix sort
   • Quick sort
   • Merge sort
   • Heap sort

Course Outcome:

After completion of this course students shall be able to-

1) Develop solutions for a range of problems using procedure oriented / object-oriented programming.	
2) Choose the appropriate data structure and algorithm design method for a specified application.	
3) Apply practical knowledge on the applications of data structures.	

Course Code: **CA LAB-VII(A):**	**LAB on Machine Learning**	*Total Marks: 50*

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^2etc.
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.

3. Apply appropriate data sets to the Machine Learning algorithms. 4. Identify and apply Machine Learning algorithms to solve real world problems.

Course Code: **CA LAB-VII(B)**	**LAB On Digital Image Processing and Computer Vision**	Total Marks: 50

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 burring
   (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.

Course Code: **CA LAB-VIII(A)**	**LAB on Advanced Java (Technologies)**	*Total Marks:* **50**

Course Objectives:
1) Study Step-by-Step procedure for building the project in java from ground up by using IDE.
2) Develop application using collection framework, RMI technology, JavaBeans and EJB
3) Develop Web Applications using advanced Java technology Servlets , JSP, Strut and Hibernate

1] Write java program(s) that demonstrates generic programming.
2] Write a Java program(s) that demonstrates the use of Collection Classes (Collection framework).
3] Write a Java program(s) that demonstrates the use of RMI technology.
4] Write a Java program(s) that demonstrates Java Bean.
5] Write a Java program(s) that demonstrates EJB.
6] Write a Java program(s) that demonstrates use of Servlets.
7] Write a Java program(s) that demonstrates use of JSP technology.
8] Implement the dynamic web application(s) to demonstrate use of struts.
9] Implement the dynamic web application(s) to demonstrate use of Hibernate.

Course Outcome:
After completion of this course students shall be able to-
1) Step-by-Step procedure for building the project from ground up by using IDE.
2) Create dynamic web application to utilize the JavaBeans and EJBs reusable components
3) Create web application using servlets, JSP, Strut and Hibernate technologies.

Course Code: **CA Lab-VIII(B)**	**LAB on Python programming**	*Total Marks:* **50**

Course Objectives:
1) To acquire programming skills in core Python.
2) To develop the skill of designing Graphical user Interfaces in Python
3) To develop the ability to write file handling, exception handling and modular programming applications in Python.

1. Develop programs to understand the control structures of python
2. Develop programs to learn different types of structures (list, dictionary, tuples) in python
3. Develop programs to learn concept of functions scoping, recursion and list mutability.
4. Develop programs to understand object oriented programming using python.
5. Develop programs for data structure algorithms using python – searching, sorting and hash tables.
6. Develop programs to learn regular expressions using python.
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
8. Demonstrate the concept of String-Based Exceptions, Class-Based Exceptions and Nesting Exception handlers.
9. Demonstrate implementation of the Anonymous Function Lambda.
10. Demonstrate implementation functional programming tools such as filter and reduce
11. Demonstrate the Module Creation, Module usage.
12. Demonstrate image insertion in python.
13. Demonstrate use of DataFrame method and use of .csv files.
14. Develop programs to learn GUI programming using Tkinter.

15. Create a simple web application using Flask.

16. Create Simple Django Framework.

17. Demonstrate Database connectivity using MySql.

Course Outcome:

After completion of this course students shall be able to-
1) Demonstrate use and working of various data types, control structures, files, exceptional handling etc.
2) Create, configure and make use of modules.
3) Develop console based and GUI applications (both procedural/object oriented) to solve different problems using python programming.

Semester-III

Course Code: **CA-301**	**Compiler Construction**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
    1) To cover the major topics in compiler design with emphasis on solving the problems encountered in designing a compiler regardless of the source language or the target machine.

Unit-I	**[05]**	Max Marks:**10**

Introduction to Compilation
Compiler Basics, Issues in Compilation, Phases of Compilation: the Analysis – Synthesis Model, Compiler Construction Tools.

Unit-II	**[06]**	Max Marks:**15**

Designing a Lexical Analyzer
Role of Lexical Analysis, Input Buffering, Specification of Tokens, Recognition of Tokens, Finite automata, Conversion from regular expression to NFA, Deterministic finite automata, Conversion from NFA to DFA, Minimization of DFA, Creating Lexical Analyzer with LEX.

Unit-III	**[15]**	Max Marks:**30**

Designing Syntax Analyzer
Role of Syntax Analyzer, Classification of parsers, **Top-Down Parsing:** Introduction, Problems in top-down parsing, Recursive Parsing, Problems in Recursive Procedures, Predictive Parsing, Error Handling in Predictive Parsers, **Bottom Up Parsing**: Shift Reduce Parser, Actions of shift reduce parser, Construction of parse tree, Operator Precedence Parsing, Components of operator precedence parsers, Parsing action, Construction of operator precedence parsers, Error reporting and recovery in operator precedence Parsers, Advantages and disadvantages of operator precedence Parsing. LR Parsing: Simple LR parser, LR (1) parser, LALR parser.

Unit-IV	**[10]**	Max Marks:**15**

Intermediate Code Generation
Need For Intermediate Code Generation, Intermediate Forms: Polish Notation, Quadruples, Triples, Indirect Triples & Blocks.

Unit-V	**[10]**	Max Marks:**15**

Code Optimization
Introduction, need for code optimization, Classification of code optimization techniques: Optimization techniques that work on machine code, Optimization techniques that work on intermediate forms of source code i.e. Optimization with in Basic Blocks: Folding, Redundant operation elimination, Optimization with in Loop: Strength Reduction, Dead code elimination, Moving operation within block out of block.

Unit-VI	**[10]**	Max Marks:**15**

Symbol Table Organization
Introduction, Methods of organizing a symbol table: Unsorted, sorted symbol tables, binary search, hashing, its advantages, disadvantages, Collision, Collision resolution techniques: Rehashing, Chaining.

References:
1. Aho A.V., R. Sethi and J.D. Ullman. Compiler Principle, Techniques and Tools: Addison Wesley, ISBN 0-321-48681-1.
2. Barret, Couch. Compiler Construction Theory and Practice: Computer Science series, Asian Student Ed, ISBN 978-0574213358
3. Dhamdhere D.M. Compiler Construction Principle and Practice: McMillan India, ISBN 9780333904060
4. Gres D. Compiler Construction for Digital Computer: Wiley, ISBN 047132776X.
5. David Galles (2009). Modern Compiler Design: Pearson Education, ISBN 9788131709412

Course Outcome:
After completion of this course students shall be able to-
1) Understand the basic structure of compiler, concepts and terminology in programming languages.
2) Explain lexical analysis, finite state techniques, scanner generator, parsing, kinds of parsers, designing lexical analyzer, scanner and parsers, principal ideas with intermediate code generation, optimizations.
3) Understanding of all concepts is essential to design compiler in general for programming languages.

Course Code: **CA-302**	**Design and Analysis of Algorithms**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
1) To understand Basics of algorithms, design techniques and analyze the performance.
2) To learn Searching and traversal algorithms for graphs.
3) To understand Nondeterministic algorithms and NP class of problem.

Unit-I	**[06]**	Max Marks:**08**

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,

Unit-II	**[06]**	Max Marks:**12**

Tree and Graph Representations, Binary Trees Basics, Heaps And Heap Sort, Sets And Disjoint Set Union And Find.

Unit-III	**[12]**	Max Marks:**16**

Divide And Conquer: General Method, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Strassen's Matrix Multiplication.

Unit-IV	**[08]**	Max Marks:**16**

Greedy Method: General Method, Optimal Storage on Tapes, Knapsack Problem, Huffman Code, Minimum-Cost Spanning Trees, Single-Source Shortest Paths.

Unit-V	**[08]**	Max Marks:**14**

Dynamic Programming: General Method, All-Pair Shortest Path, Matrix Chain Multiplication, Longest Common Sub Sequence, 0/1knapsack, Flow Shop Scheduling

Unit-VI	**[08]**	Max Marks:**10**

Basic Search and Traversal Techniques: Breadth First Search and Traversal, Depth First Search And Traversal, Spanning Trees.

Unit-VII	**[06]**	Max Marks:**12**

Backtracking: General Method, Constrains, 8-Queens Problem Graph Coloring

Unit-VIII	**[06]**	Max Marks:**12**

NP-Hard and NP-Complete Problems: 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

References:

5. Horowitz E. and Sahni S. "Fundamentals of computer Algorithms" Galgotia publications.
6. Horowitz E., Sahni S. and Rajshekaran S,Computer Algorithms, Computer Science Press.
7. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani(2006), Algorithms. McGraw-Hill publications. ISBN 9780073523408
8. 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-
1) Analyze the asymptotic performance of algorithms and write rigorous correctness proofs for algorithms.
2) Design and analyze divide-and-conquer, greedy and dynamic-programming based algorithms.
3) Model problems using backtracking, classify nondeterministic polynomial time algorithms.

Course Code: **CA-303**	**High Performance Computing Paradigms and Applications**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:

   1) Comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications

   2) To expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

   4) To provide knowledge of cloud concepts capabilities across the various cloud service models which includes IaaS, PaaS and SaaS; and learn how to use Cloud Services, implementation of Virtualization, Task Scheduling algorithms.

Unit-I	**[06]**	Max Marks:**20**

Basics of Cloud Computing: Introduction to Distributed Systems, Single System Image, Naming and Synchronization Communication in DS, Load Balancing, Process Migration, Fault Tolerant Systems. Introduction to Grid and Cluster Computing.

Unit-II	**[06]**	Max Marks:**15**

Introduction to Cloud Computing: Roots of Cloud Computing, Layers and Types of Clouds, Features of a Cloud, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers, and Challenges and Opportunities

Unit-III	**[06]**	Max Marks:**10**

Virtualization and Resource Provisioning in Clouds: Introduction and Inspiration, Virtual Machines (VM), VM Provisioning and Manageability, VM Migration Services, VM Provisioning in the Cloud Context, and Future Research Directions.

Unit-IV	**[10]**	Max Marks:**10**

Cloud Computing Architecture: Cloud Benefits and Challenges, Market-Oriented Cloud Architecture, SLA-oriented Resource Allocation, Global Cloud Exchange; Emerging Cloud Platforms, Federation of Clouds

Unit-V	**[12]**	Max Marks:**15**

Programming Enterprise Clouds using Aneka: Introduction, Aneka Architecture, Aneka Deployment, Parallel Programming Models, Thread Programming using Aneka, Task Programming

using Aneka, and MapReduce Programming using Aneka, Parallel Algorithms, Parallel Data mining, Parallel Mandelbrot.		
Unit-VI	**[12]**	Max Marks:**15**
Advanced Topics and Cloud Applications: Integration of Private and Public Clouds, Cloud Best Practices, GrepTheWeb on Amazon Cloud, ECG Data Analysis on Cloud using Aneka, Hosting Massively Multiplayer Games on Cloud.		
Unit-VII	**[08]**	Max Marks:**15**
Practical Related Topics: Topics related to Practical hands on will be added later on.		
Unit-I	**[06]**	Max Marks:**20**
Basics of Cloud Computing: Introduction to Distributed Systems, Single System Image, Naming and Synchronization Communication in DS, Load Balancing, Process Migration, Fault Tolerant Systems. Introduction to Grid and Cluster Computing.		
References		
1. A. S Tanenbaum and M.V. Steen(2007), Distributed Systems: Principles and Paradigms, Second Edition, Pearson Prentice Hall Publication, 2007, ISBN:0-13-239227-5		
2. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi(2013), Mastering Cloud Computing, Tata McGraw Hill, New Delhi, India, 2013 **ISBN**-13: 978-1-25-902995-0		
3. Rajkumar Buyya, James Broberg , Andrzej M. Goscinski(2011), Cloud Computing: Principles and Paradigms, Wiley India Publication, 2011, ISBN: 978-0-470-88799-8		
4. Toby Velte, Anthony Velte, Robert Elsenpeter (2009), Cloud Computing: A Practical Approach, Tata McGraw Hill Publication,2009, ISBN 0071626948 / 9780071626941		
5. Barrie Sosinsky (2011), Cloud Computing Bible, Wiley Publishing India Pvt Ltd.,2011, ISBN: 978-0-470-90356-8		
Course Outcome: After completion of this course students shall be able to- 1) Analyze the Cloud computing setup with its vulnerabilities and applications using different architectures. 2) Design suitable Virtualization concept, Cloud Resource Management. 3) Assess cloud Storage systems and Cloud security, the risks involved, its impact and develop cloud application.		

Course Code: **CA-304**(A)	**Natural Language Processing**	*Clock Hours:* **60** *Total Marks:* **100**
Course Objectives: 1) 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. 2) 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. 3) 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.		
Unit-I	**[08]**	Max Marks:**12**
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).

Unit-II	[12]	Max Marks:**16**

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.

Unit-III	[12]	Max Marks:**18**

Word Classes ad Part-of-Speech tagging(POS), survey of POS tagsets, Rule based approaches (ENGTOWL), Stochastic approaches(Probabilistic, N-gram and HMM), TBL morphology, unknown word handling, evaluation metrics: Precision/Recall/F-measure, error analysis.

Unit-IV	[15]	Max Marks:**22**

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.

Unit-V	[15]	Max Marks:**22**

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.

References:
1. Indurkhya, N., & 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
2. Martin, J. H., & Jurafsky, D.(2013), *Speech and Language Processing,* Pearson Education India; 2 edition, ISBN-10: 9332518416, ISBN-13: 978-9332518414
3. Manning, Christopher and Heinrich, Schutze(1999), Foundations of Statistical Natural Language Processing", MIT Press, ISBN-10: 0262133601, ISBN-13: 978-0262133609.
4. Akshar Bharati, Chaitanya, V., Kulkarni, A., & Sangal, R. (July 1997). *Machine translation in Stages* (Vol. 10 no. 3). Mumbai: NCST, Mumbai.
5. 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.
6. Steven Bird, Edward Loper (2016),Natural Language Processing With Python, Ed. 2, O'Reilly Media,ISBN 1491913428, 9781491913420

Auxiliary Resources:

Web Links
1. https://see.stanford.edu/Course/CS224N
2. https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html
3. https://stp.lingfil.uu.se/~nivre/docs/ACLslides.pdf

Video Links
1. http://www.nptelvideos.in/2012/11/natural-language-processing.html
2. https://www.youtube.com/playlist?list=PL6397E4B26D00A269

Course Outcome:
After completion of this course students shall be able to-
1) Understand issues and challenges in Natural Language Processing and NLP applications and their relevance in the classical and modern context.
2) 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.

Course Code: **CA-304 (B)**	**Artificial Intelligence in Practice with Python**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
The objectives of the course are:
1) How to make informed decisions about the type of algorithms you need to use and implementation of these algorithms for solving AI problems.
2) Use of artificial intelligence techniques for solving practical real-life problems.
3) To build real-world artificial intelligence applications.

Unit-I	**[03]**	**Max Marks:06**

Fundamental Use Cases for AI
What is AI?, Why AI?, Branches of AI, Building Intelligent Agents, Representative AI Use Cases, Digital Personal Assistant and Chatbots, Shipping and Warehouse Management, Human Health, Knowledge Search, Recommender Systems, The Smart Home, Gaming, Movie Making, Data Cleansing and Transformation.

Unit-II	**[04]**	**Max Marks:10**

Machine Learning Pipelines
What is a Machine Learning Pipeline? Problem Definition, Data Ingestion, Data Preparation, Data Segregation, Model Training.

Unit-III	**[05]**	**Max Marks:08**

Feature Selection and Feature Engineering
Feature Selection, Feature Engineering, Outlier Management, One-hot Encoding, Log Transform, Scaling, Date Manipulation.

Unit-IV	**[06]**	**Max Marks:10**

Classification and Regression Using Supervised Learning
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.

Unit-V	**[05]**	**Max Marks:08**

Predictive Analytics
Decision Trees, Ensemble Learning, Random Forests, Dealing with Class Imbalance, Finding Optimal Training Parameters with Grid Search, Computing Relative Feature Importance, Case Study: Use Extremely Random Forest Regressor for Application like Predicting Traffic.

Unit-VI	**[06]**	**Max Marks:08**

Detecting Patterns with Unsupervised Learning

Unsupervised Learning, Clustering Data with K-Means Algorithm, Estimating Number of Clusters with the Mean Shift Algorithm, Estimating Quality of Clustering with Silhouette Scores, Gaussian Matrix Models, Finding Subgroups in Stock Market Using the Affinity, Propagation Model, Segmenting the Market based on Shopping Patterns.

Unit-VII	[05]	Max Marks:08

Building Recommender Systems

Extracting the nearest Neighbors, Building K-Nearest Neighbors Classifier, Commuting Similarity Scores, Finding Similar Users Using Collaborative Filtering, Case Study: Building Movie Recommender System.

Unit-VIII	[07]	Max Marks:10

AI on Cloud

Why are Companies migrating to Cloud?, Top Cloud Providers, Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).

Unit-IX	[05]	Max Marks:08

Building Speech Recognizer

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.

Unit-X	[06]	Max Marks:08

Building NLP Tools

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.

Unit-XI	[04]	Max Marks:08

Chatbots

Chatbots today, Concepts, A well-architected chatbot, Platforms, Creating a chatbot,

Unit-XII	[04]	Max Marks:08

AI and Big Data

Crawling, Indexing, Ranking, Worldwide datacenters, Distributed lookups, Custom softwares, The three V's of Big data, Big data and machine Learning - Apache Hadoop, Apache Spark, Apache Impala, NoSQL databases - Apache Cassandra, MangoDB, Redis, Neo4j.

References:

1] Python Machine Learning, Sebastian Raschka, Vahid Mirjalili, Packt Publishing, ISBN: 978-1-78995-575-0

2] AI Crash Course, Hadelin de Ponteves, Packt Publishing, ISBN: 978-1-83864-535-9

3] Artificial Intelligence with Python, Prateek Joshi, Packt Publishing, Packt Publishing, ISBN: 9781786464392

Course Outcome:

After completion of this course students shall be able to-

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

Course Code: **CA-304(C)**	**Data Analytics**	*Clock Hours:* **60** *Total Marks:* **100**
Course Objectives:		

1) This course will cover fundamental algorithms/techniques used in data analytics and will provide exposure to theory as well as practical systems and software used in data analytics.		
2) The statistical foundations will be covered first, followed by various machine learning and data mining algorithms.		
3) 3. Technological aspects like data management (Hadoop), scalable computation (MapReduce) and visualization will also be covered. In summary, this course		
Unit-I	**[10]**	Max Marks:08
Data Definitions and Analysis Techniques		
• Elements, Variables, and Data categorization • Levels of Measurement • Data management and indexing • Introduction to statistical learning and R-Programming		
Unit-II	[10]	Max Marks:10
Descriptive Statistics		
• Measures of central tendency • Measures of location of dispersions • Practice and analysis with R		
Unit-III	[13]	Max Marks:15
Basic analysis techniques		
• Statistical hypothesis generation and testing • Chi-Square test • t-Test • Analysis of variance • Correlation analysis • Maximum likelihood test • Practice and analysis with R		
Unit-IV	**[15]**	Max Marks:17
Data analysis techniques		
• Regression analysis • Classification techniques • Clustering • Association rules analysis • Practice and analysis with R		
Unit-V	**[12]**	Max Marks:10
Case studies and projects		
• Understanding business scenarios • Feature engineering and visualization • Scalable and parallel computing with Hadoop and Map-Reduce • Sensitivity Analysis		
References:		
1) Probability & Statistics for Engineers & Scientists (9th Edn.), Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying Ye, Prentice Hall Inc. 2) The Elements of Statistical Learning, Data Mining, Inference, and Prediction (2nd Edn.), Trevor Hastie Robert Tibshirani Jerome Friedman, Springer, 2014 3) An Introduction to Statistical Learning: with Applications in R, G James, D. Witten, T Hastie, and R. Tibshirani, Springer, 2013		

4) Software for Data Analysis: Programming with R (Statistics and Computing), John M. Chambers, Springer
5) Mining Massive Data Sets, A. Rajaraman and J. Ullman, Cambridge University Press, 2012
6) Advances in Complex Data Modeling and Computational Methods in Statistics, Anna Maria Paganoni and Piercesare Secchi, Springer, 2013
7) Data Mining and Analysis, Mohammed J. Zaki, Wagner Meira, Cambridge, 2012
8) Hadoop: The Definitive Guide (2nd Edn.) by Tom White, O'Reilly, 2014
9) MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems, Donald Miner, Adam Shook, O'Reilly, 2014
10) Beginning R: The Statistical Programming Language, Mark Gardener, Wiley, 2013

Course Outcome:

After completion of this course students shall be able to-
1) Find a meaningful pattern in data; graphically interpret data.
2) Implement the analytic algorithms.
3) Handle large scale analytics projects from various domains; Develop intelligent decision support systems.

Course Code: **CA-305(A)**	**Mobile Application Development (Android Programming)**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
1) Understand basics of mobile application development and get introduced Android platform and its architecture.
2) To learn activity creation and Android UI designing.
3) 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.

Unit- I	**[05]**	Max Marks:**08**

Mobile Application Development: Introduction to handheld devices (Palm, Pocket Pc, Symbian OS smart phones, MS windows based smart phones, BlackBerry, iphone etc.), features of handheld devices, Device Applications Vs Desktop application, overview of application development platforms (OS-Palm OS, Symbian, BlackBerry, Windows CE, OS for iphone, Android), Programming Languages (C/C++, JAVA), IDE tools. Comparison of Android with other Mobile OS. Comparative study of all versions of Android.
[Note: The unit is to be thought with respect to current scenario of Mobile Development hence above contents may change]

Unit-II	**[06]**	Max Marks:**16**

Hello, Android and Installations: Background, What is android and what isn't, Open Mobile Development Platform, Native Android Applications, Android SDK Features, Introducing the Open Handset Alliance, What Does Android Run On? Why Develop for Android?, Introducing the Development Framework. What Comes in the Box, Developing for Android, Developing for Mobile Devices, Android Development Tools as per current version, Installations, Emulator.

Unit-III	**[16]**	Max Marks:**24**

Creating Applications, activities and User Interfaces: What Makes an Android Application?, Introducing the Application Manifest. Using the Manifest Editor, The Android Application Life Cycle. 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.		
Unit-IV	**[16]**	Max Marks:**24**
Intents, Broadcast Receivers, and the Internet: Using Intents to Launch Activities, Explicitly Starting New Activities, Implicit Intents and Late Runtime Binding, Returning Results from Activities, Native Android Actions, Linkify, Native Linkify Link Types, Creating Custom Link Strings, Match Filter, Transform Filter, Intents to Broadcast Events, Listening using Broadcast Receivers, Ordered Intents, Sticky Intents, Local Broadcast Manager, Pending Intents, Intent Filters and Broadcast Receivers, Intent Filters to Service Implicit Intents, Resolving Intent Filters, Intents Received Within an Activity, Intent Filters for Plug-Ins and Extensibility, New Actions from Third-Party Intent Receivers, Incorporating Anonymous Actions as Menu Items, Listening for Native Broadcast Intents, Device State Changes Using Broadcast Intents, Connecting to an Internet Resource, Parsing XML Using the XML Pull Parser, Download Manager, Internet Services, Connecting to Google App Engine, Introducing Dialogs and Action Bars, Creating and Using Menus.		
Unit-V	**[06]**	Max Marks:**12**
Working in the Background: Creating and Controlling Services, Binding Services to Activities, Creating Foreground Services, Using AsyncTask to Run Asynchronous Tasks, Intent Service, Loaders, Creating, Setting, and Canceling Alarms, Setting Repeating Alarms.		
Unit-VI	**[06]**	Max Marks:**08**
Files, Saving State, and Preferences: Application Data Saving, Shared Preferences: Creating, Saving, Retrieving, Preference Framework, Preference Activity, Application Instance State, Static Files as Resources, File System.		
Unit-VII	**[05]**	Max Marks:**08**
Databases and Content Providers: Android Databases, SQLite, Content Values and Cursors, data manipulation using SQLite, Content Providers, Create Content Providers use for data manipulation, adding searching in application.		

References:
1. Reto Meier. Professional Android Application Development, Wrox Publications ISBN: 978-0-470-34471-2.
2. 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.

Auxiliary   Resources:
https://developer.android.com/index.html

Course Outcome:
After completion of this course students shall be able to-
1) Compare android with other smartphone OS and desktop OS; Able to understand software stack of android OS.
2) Understand Activity lifecycle, UI management, use Intent, Broadcast receivers and Internet services.
3) Effectively use SQLite Database and content providers, multimedia, camera and Location based services in Android Application.

Course Code: **CA-305(B)**	**Microsoft .Net Technologies**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:
1) To learn .Net Framework and creating ASP.Net web applications using standard .net controls.
2) Develop database applications using ADO.Net and Connecting to data sources and managing them.
3) Develop a data driven web application.
4) Use Web Services and develop simple and complex applications using .Net framework
5) Maintain session and controls related information for user used in multi-user web applications

Unit-I	**[05]**	**Max Marks:08**

Desktop Computing vs. Internet Computing, Internet computing infrastructure, Client side scripting vs. Server Side Scripting technologies, Web Server basics and configuration: IIS, Apache etc., Web site hosting basics, Web Publishing, HTML, introduction to .NET framework, Features of .NET framework: CTS,CLS,CLR.

Unit-II	**[10]**	Max Marks:**10**

.NET technologies,languages'C#.NET,VB.NET, basics of ASP.NET page framework, Visual studio .NET IDE, Page Life Cycle, PostBack, Viewstate, Page directives, ASP.Net page execution cycle, HTTP Pipeline, HTTP Application, HTTP Request, HTTP Response classes, HTTP Modules and HTTP Handlers, State Management, Role of Global.asax, Application configuration using web.config file

Unit-III	**[12]**	Max Marks:**20**

ASP.NET Control hierarchy, HTML Server Controls, Web Server Controls, User and Server controls, Validation Controls, List bound controls:dropdown lists, list boxes, Repeater, DataList, Data Grid, DataGridView, FormsView controls, Data binding to List Bound Controls, Templating and Styling of ASP.NET server controls

Unit-IV	**[10]**	Max Marks:**22**

Web Page Designing principles, CSS anatomy, Anatomy of Master Pages, nesting master pages, Site map file, Web site Navigation controls, properties:TreeView, Sitemap Path, Menu, Other Navigation methods: Response.Redirect(), Server.Transfer().

Unit-V	**[10]**	Max Marks:**15**

Personalization through Profiles, Themes/Skins, Web Site security basics: authentication modes:Windows,Forms,passport, authorization, roles/Membership, access rules, login controls,Web services: working, anatomy, hosting

Unit-VI	**[13]**	Max Marks:**25**

Database technology: ADO.NET, Anatomy/architecture of ADO.NET, working with Connection, Command, Data Adaptor, DataReader, DataSet, DataTable objects, Editing data in Data Tables, concurrency control. Introduction to MVC, Data Reports

References:
1. Richard Anderson, Brian Francis, Alex Homer, Rob Howard, David Sussman, Karli Watson(2002), Professional ASP.NET 1.0, Special Edition, Wrox Press Ltd., 2002, ISBN 1-861007-0-3-5.
2. Chris Hart, John Kauffman, Dave Sussman, and Chris Ullman(2006), Beginning ASP.NET 2.0, Wiley Publishing, Inc., 2006, ISBN-13: 978-0-7645-8850-1, ISBN-10: 0-7645-8850-8.
3. Beginning ASP.NET 4: in C# and VB**,** Imar Spaanjaars, Wiley Publishing, Inc 2010., ISBN: 978-0-470-50221-1
4. Bill Evjen, Scott Hanselman, Devin Rader (2008), Professional ASP .NET 3.5 in C# and VB, Wiley Publishing Inc.,2008 ISBN:978-0-470-18757-9.

5. Dino Esposito (2008), Programming Microsoft ASP.NET 3.5, Second Edition, Microsoft Press, 2008, ISBN-10: 0735625271, ISBN-13: 978-0735625273

Auxiliary Resources:
   a. Website URLs
https://www.asp.net/
http://asp.net-tutorials.com/
   b. Video Links
   1. https://www.asp.net/web-forms/videos
   2. https://www.youtube.com/playlist?list=PL6n9fhu94yhXQS_p1i-HLIftB9Y7Vnxlo&feature=view_all

Course Outcome:
After completion of this course students shall be able to-
1. Design Web applications / Website using ASP.NET.
2. Use ASP.NET controls in web applications
3. Debug and deploy ASP.NET web applications.
4. Create database driven ASP.NET web applications and web services.

Course Code: **CA-305(C)**	**Ruby on Rails**	*Clock Hours:* **60** *Total Marks:* **100**

Course Objectives:

Unit-I	**[08]**	Max Marks:**12**

Version control: Github, Branching Strategies
Ruby Installation and Basics: Ruby/Rails Installation, 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

Unit-II	**[10]**	Max Marks:**15**

Ruby core: 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

Unit-III	**[04]**	Max Marks:**06**

Ruby I/O: File I/O, File inquiries, Directories, Navigation through Directories

Unit-IV	**[10]**	Max Marks:**15**

Ruby Advanced: Ruby OOPs concepts, Database Access, Web Application without framework, Sending email, Ruby - XML, XSLT and XPath

Unit-V	**[10]**	Max Marks:**15**

Ruby Advanced: Ruby TK (GUI for Ruby), Ruby – Multithreading Built in Functions, Variables, Constants, Ruby associated tools, Ruby - XML, XSLT and XPath

Unit-VI	**[18]**	Max Marks:**27**

Rails Basics: 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: 1. Yukihiro Matsumoto (2008), The Ruby Programming Language, Shroff; First edition, 2008, ISBN-10: 8184044925, ISBN-13: 978-8184044928 2. Michael Fitzgerald, Learning Ruby, Published by O'Reilly Media, Inc., May 2007, ISBN-10: 8184043341, ISBN-13: 978-8184043341 3. Rails AntiPatterns, Wesley Professional Ruby Series, 1st edition, 2010, ISBN-10: 0321604814, ISBN-13: 978-0321604811 4. 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-
   1) 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.
   2) Use the Ruby TK (GUI for Ruby).
   3) Design web applications using Rails framework.

Course Code: **CA Lab-IX**	**LAB on Design and Analysis of Algorithms**	*Total Marks:* **50**
Course Objectives: 1) To convert the algorithms to code, measure the complexities at run time and modify the algorithms for efficiency. 2) To debug and test the programs. 3) To conclude using profile of outcomes.		

Laboratory Requirements:
OS: Windows/Linux,
Programming Language: C++/Java/C#

8. Write a program for creating max./min. heap using INSERT.
9. Write a program for creating max./min. heap using ADJUST/HEAPIFY.
10. Write a program to implement union and find operation.
11. Write a program to find minimum and maximum form a given array.
12. Write a program for searching element form given array using binary search for n=1000,2000,3000 find exact time of execution.
13. Write a program for sorting given array in ascending/descending order with n=1000,2000,3000 find exact time of execution using
    - Heap sort
    - Merge sort
    - Quick sort
14. Write a program for matrix multiplication using Strassen's matrix multiplication.
15. Write a program to find solution of Knapsack instant.
16. Write a program to find shortest path using single source shortest path.
17. Write a program to find Minimum-Cost Spanning Trees (Prim's & Kruskal's Algorithm).
18. Write a program to find shortest path using all pair path.
19. Write a program to find longest common subsequence.

20. Write a program to implement breadth first and depth first search.
21. Write a program to implement breadth first and depth first traversal.
22. Write a program to find all solutions for 8-queen problem using backtracking.

Course Outcome:
After completion of this course students shall be able to-
1) Construct logic for the algorithms designed using designing techniques.
2) Posterior analysis of the algorithms.
3) Debug, test and profile the algorithms, modify to improve performance of the algorithms.

Course Code: **CA LAB-X**	**Lab on High Performance Computing Paradigms and Applications**	*Total Marks:* **50**

Course Objectives:
1) Study configurations of cloud infrastructure.
2) Analyze and monitor the cloud.
3) Handle and backup real time warehouse data.

1. Study and do the Configuration of CLoudSim. Also execute & check the performance of existing algorithms.
2. Install a Cloud Analyst and Integrate with Eclipse/Netbeans. Monitor the performance of an Existing Algorithms.
3. Modify or propose a new load balancing algorithm compatible with Cloud Analyst.
4. Integrating GoogleApp Engine API's in Eclipse and develop an application in Java/Python on the top of Google Cloud.
5. Make the registration groupwise on Google and register your application by using google application-ID
6. Creating a Warehouse Application in SalesForce.com.
7. Creating an Application in SalesForce.com using Apex programming Language.
8. Implementation of SOAP Web services in C#/JAVA Applications.
9. Implementation of Para-Virtualization using VM Ware's Workstation/ Oracle's Virtual Box and Guest O.S.
10. Installation and Configuration of Hadoop.
11. Create an application (Ex: Word Count) using Hadoop Map/Reduce.
12. Case Study: PAAS(Facebook, Google App Engine)
13. Case Study: Amazon Web Services.

Course Outcome:
After completion of this course students shall be able to-
1) Configure cloud infrastructure.
2) Monitor load on cloud, balance load by analyzing.
3) Work with real time cloud solutions.

Course Code: **CA LAB XI(A)**	**Lab on Natural Language Processing**	*Total Marks: 50*

Course Objectives:
The objectives of the course are:
1) Course provides knowledge of installation and use of NLTK in python.
2) Course provides knowledge of implementation of text files processing operations and Regular Expressions in NLP

3) Course provide knowledge of implementation of dependency parser, porter stemmer, Morphology, PoS Tagging

1. Install NLTK and perform basic preprocessing steps of NLP like tokenization, stemming, lemmatization, chunking etc using NLTK in python.
2. Write a program to perform text files statistical operation like count number of lines in files, number of words in file.
3. Working with PDF files in Python like Extracting text from PDF, Rotating PDF pages, Merging PDFs, Splitting PDF, Adding watermark to PDF pages
4. Write program to count number of articles (a, an, the) in file.
5. Write a program to perform tokenization and filtering stopwords in file.
6. Write a program which makes use of basics in regular expressions like /a*/, /a+/, /a? /, /[ˆA-Z]/, /[ˆSs]/, etc.
7. Write a program for minimum edit distance algorithm.
8. 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.
9. Write a program to demonstrate use of porter stemmer in python.
10. Write a program to demonstrate use of dependency parser.
11. Write a program to demonstrate use of NP and VP chunker.
12. Write a program for Tagging Sentences which takes input as sentence and performs PoS Tagging.
13. Write a program for bigram formation from given list.

References:

1] Indurkhya, N., &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
2] Martin, J. H., &Jurafsky, D.(2013), Speech and Language Processing, Pearson Education India; 2 edition, ISBN-10: 9332518416, ISBN-13: 978-9332518414
3] Steven Bird, Edward Loper (2016),Natural Language Processing With Python, Ed. 2, O'Reilly Media,ISBN 1491913428, 9781491913420

Course Outcome:

After completion of this course students shall be able to-
1. idea about installation and use of NLTK in python.
2. understanding of implementation of text files procesing operation and Regular Expressions in NLP
3. Knowledge of implementation of dependency parser, porter stemmer, Morphology, PoS Tagging and other NLP applications

Course Code: **CA LAB-XI (B)**	**LAB on AI Practice using Python**	*Total Marks:* **50**

Course Objectives:

1) To explore most common artificial intelligence (AI) use cases.
2) To implement various new artificial intelligence techniques.
3) To create real-world AI application/s using above AI technique/s.

1. Installation of Python on Windows/Ubantu, Installing Packages, Loading data.
2. Data Preparation using techniques like Data Cleansing, Filtration, Aggregation etc
3. Handling missing values, Feature Scaling, Inconsistent values in the given dataset.
4. Feature selection using techniques like univariate selection correlation heatmaps, Wrapper-based methods, Filter-based methods.

5. Feature engineering using techniques like Outlier management, One-hot encoding, Log transform.
6. Implement Logistic regression classifier.
7. Implement Naïve Bayes classifier.
8. Use confusion matrixes to describe performance of a classifier.
9. Implement classifier using Support Vector Machines.
10. Build a decision tree classifier and evaluate performance of a classifier by printing classification report.
11. Build random forest and extremely random forest classifiers and analyze the output.
12. Implement K-Means algorithm for clustering.
13. Build K-nearest classifier
14. Visualizing audio signals.
15. Transform audio signals to the frequency domain.
16. Generate audio signals.
17. Installation of NLTK and tokenizing text data.
18. Converting words to their base forms using stemming, lemmatization.
19. Extracting the frequency of terms using Bag of Words model.

Course Outcome:
After completion of this course students shall be able to-
  1) Use most common artificial intelligence (AI) use cases in developing AI applications.
  2) Apply various new artificial intelligence techniques in developing AI applications.
  3) Create real-world AI application/s using above AI technique/s.

Course Code: **CA LAB-XI(C)**	**Lab on Data Analytics**	*Total Marks:* **50**

Course Objectives:
  1) Learn Data Science concepts of R and functioning of R
  2) Understand Exploratory Data Analytics
  3) Learn to program various analysis techniques

1. Write program for Creating and Manipulating R Objects in R – Vectors, Matrices, Arrays, Data Frames and Lists.
2. Write program to demonstrate Loops & Vectorization Missing Values.
3. Demonstrate Importing and exporting data.
4. Write program for Validating & Exploring Data Manipulations (Summarizing, Sorting, Sub-setting, Merging, joining)
5. Write program to implement the following analysis techniques using R
   - Statistical hypothesis generation and testing
   - Chi-Square test
   - 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 R programming constructs.
2) Manipulate data using R.
3) Write code for various data analysis techniques.

Course Code: **CA LAB-XII(A)**	**LAB on Android Programming**	*Total Marks:* **50**

Course Objectives:
1) To gain knowledge of installing Android Studio and Cross Platform Integrated Development Environment;
2) To learn designing of User Interface and Layouts for Android App, intents to broadcast data within and between Applications.
3) To use Content providers and Handle Databases using SQLite.

Assignments:
1. Demonstrate string manipulation by displaying at the middle of the screen in the red color with white background with change in fonts & styles of text.
2. Write a program to populate resources (res>>value folder). Show resource on changing selection of the resources.
3. 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.
4. Write a program for android application to demonstrate android life cycle stages.
5. Create the application that will change color of screen based on selected option from the menu.
6. Write an android application that takes input from user and shows messages on screen.
7. Create foreground application that will display toast (Message) on specific interval time.
8. Demonstrate use of intents for any 3 native intents.
9. Create the android application that will read phonebook contact using content providers and display in list on selecting specific contact makes a call to selected contact.
10. Develop android application to take a picture using native application.
11. Use fragments to develop UI. Demonstrate use of fragments.
12. Create the android application that will send SMS using your android application.
13. Write a android background service that will open activity on specific time.
14. Demonstrate use of shared preferences.
15. Write code that will call maps using android application.
16. Develop application for database manipulation.
    Mini project: Develop an app in android.

Course Outcome:
After completion of this course students shall be able to-
1) Design and Implement User Interfaces and Layouts of Android App; Use Intents for activity and broadcasting data in Android App.
2) Design and Implement Database Application and Content Providers.
3) Develop Android App with Security features.

Course Code: **CA LAB-XII(B)**	**Lab on Microsoft .Net Technologies**	*Total Marks:* **50**

Course Objectives:
1) Students will understand Web Sites / Web applications, basics of Web hosting and working of IIS web server.

| 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. |
| 3) | Understand the fundamentals of developing modular application by using object-oriented methodologies. |

Assignments:

Demonstrate followings in IIS:
1) Creation of Virtual Directory, Home directory, Home page, hosting of website
2) Demonstrate Page Life Cycle of ASP.NET. Use important page events for your demonstration.
3) Write VB.Net/C# console applications to demonstrate: OO concepts: polymorphism, encapsulation, inheritance, interface inheritance, abstract classes/methods, overloading, overriding, collection classes, properties
4) Demonstrate concept of postback and viewstate using web form server controls of ASP.NET
5) 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.
6) Demonstrate DropDown List box, CheckButtonList, RadioButtonList controls.
7) Demonstrate Databinding using Hashtable, ArrayList, DataTable data sources.
8) Demonstrate Repeater control with the help of various templates.
9) Demonstrate paging, sorting,filtering of data in asp:DataGrid/DataGridView.
10) Demonstrate editing process in DataGrid and DataList controls. Make use of necessary templates for proper visual appearance.
11) Demonstrate State Management features of ASP.NET using sample shopping cart application.
12) Create sample website for demonstrating use of Profiles/Themes using skin files.
13) Demonstrate Master Pages and website navigation controls(sitemap path, treeview, menu) using SiteMap file.
14) Demonstrate Properties of website navigation controls.
15) Demonstrate Authorization/Authentication using Login controls and Roles/Membership/AccessRules
16) Demonstrate creation of simple/complex DataReader/DataSet Objects.
17) Demonstrate editing in DataTable objects.
18) Demonstrate Web Service hosting, access in ASP.NET

Course Outcome:

After completion of this course students shall be able to-
1) Design web site and web applications using ASP.NET
2) Debug and deploy ASP.NET web applications
3) Create database driven ASP.NET web applications and web services.

Course Code: **CA LAB-XII(C)**	**LAB on Ruby on Rails**	*Total Marks:* **50**

Course Objectives:
1) Install Ruby on Rails
2) Write programs in ruby.
3) Develop applications using rails framework.

Assignments:

1. Demonstrate a Ruby Basic program which manipulate Hash, Array, Strings. Any five methods of each container. Also use !(bang) operator.

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 cirletom@laptop:~/courses/ruby/asst$ ruby radius.rb
   Enter the radius: 2
   Area is: 12.5663708
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 for spaces or punctuation.
4. Demostrate Inheritence 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)
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 devide by zero, etc.
6. Write a Ruby script which demonstrate use of blocks, lambd &proc.
7. Write a Ruby program which show duck typing, uses respond_to? Method.
8. Write a Ruby program which access private methods/attributes outside of class.
9. 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.
10. Create a Basic Ruby on Rails web application which print "Hello World on web browser"
11. Create a Ruby on Rails web application which shows having Post Section. In which user can Insert, Edit, Delete Post, using scaffolding.
12. Create a Ruby on Rails web application with Post Model uses variuos type of server validation.
13. 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.
14. 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 nestes routes.like "/posts/2/comments"
15. Create a Ruby on Rails web application using mysql database. Post is always belongs to user and user has many posts. Without login user cann't Insert, Delete or Edit Post, can only show post using devise gem.
16. Create a Ruby Application having 3 to 4 .rb files interconnected with each other. Which demonstrate all above concept with Human readable console output.
17. Design a Ruby On Rails Web Application which deals with User, Registration Form, Validations, CSS, JavaScripts, Ajax, Associations, etc

*Course Outcomes***:**
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.

Course Code: **CA-401**   **Full Time Industrial Training**   *Total Marks:* **300**

Course Objectives:
1) To provide comprehensive learning platform to students where they can enhance their employ ability skills and become job ready along with real corporate exposure.
2) To enhance students' knowledge in a particular technology and to Increase self-confidence of students and helps in finding their own proficiency.
3) To cultivate student's leadership ability and responsibility to perform or execute the given task.

Twelve credits shall be awarded to the Industrial Training/Project course, which will commence in the IVth Semester and the final work and report will be completed at the end of IVth Semester of M. C.A. The student is expected to work on software development project. The project work should have coding part. Student will have to submit the bound project report in university prescribed format at the end of the semester. Student will have to appear for Project Viva-voce and the marks and the credits will be allotted at the end of IVth semester of M. C.A.

Course Outcomes:
After completion of this course students will:

1) Handle specialized technology and update themselves with latest changes in technological world with ability to communicate effectively.
2) Be multi-skilled IT professional with good technical knowledge, management, leadership and entrepreneurship skills.
3) Be able to identify, formulate and model problems and find engineering solution based on a systems approach.

Program Specific Outcomes for M.C.A. program

At the end of the program the graduate will be able to:

- Apply knowledge of computer science in practice to identify, critically analyze, formulate and develop computer applications using modern computing tools and techniques and will use these tools with dexterity.

- Design computing systems to meet desired needs within realistic constraints such as safety, security and applicability. These systems will function professionally with ethical responsibility as an individual as well as in multidisciplinary teams with positive attitude.

- Appreciate the importance of goal setting and recognize the need for life-long learning with good communication skills.