



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

School of Computer Sciences

MASTER OF COMPUTER APPLICATIONS (MCA)

Syllabus

[under CBCS and Academic Flexibility]

Faculty of Science and Technology

With effect from 2019-20

MASTER OF COMPUTER APPLICATIONS (MCA)

PROGRAMME OBJECTIVES (POs):

- 1) Prepare graduates to become computer professionals with comprehensive knowledge and skills to produce software for emerging requirement
- 2) Prepare graduates to become continuous learner with aptitude for teaching and research with societal focus
- 3) To prepare graduates who will achieve peer-recognition; as an individual or in a team; through demonstration of good analytical, design and implementation skills
- 4) To prepare graduates who will thrive to pursue life-long learning to fulfill their goals

PROGRAMME SPECIFIC OUTCOMES (PSOs):

MCA Programme has been designed to prepare graduates for attaining the following program outcomes:

- 1) An ability to apply knowledge of computer science in practice
- 2) An ability to identify, critically analyze, formulate and develop computer applications
- 3) An ability to select modern computing tools and techniques and use them with dexterity
- 4) An ability to design a computing system to meet desired needs within realistic constraints such as safety, security and applicability
- 5) An ability to function professionally with ethical responsibility as an individual as well as in multidisciplinary teams with positive attitude
- 6) An ability to communicate effectively
- 7) An ability to appreciate the importance of goal setting and to recognize the need for life-long learning

MASTER OF COMPUTER APPLICATIONS (MCA)

Degree Name	:	Master in Computer Applications (MCA) [Under Academic Flexibility]
Faculty	:	Science and Technology
Duration	:	03 years, Full Time course
Medium of Instruction	:	English
Pattern	:	Semester Pattern (06 semesters)
Examination Pattern	:	60% (External Assessment) + 40%(Internal Assessment)
Passing Standard	:	Separate Passing for internal as well as external assessment.
Evaluation Mode	:	CGPA
Lecture	:	Clock hour (60 minutes)

Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon
School of Computer Sciences
Syllabus under CBCS for Master of Computer Applications (MCA)
Course Structure (w.e.f. 2019-20)

COURSE STRUCTURE WITH CREDIT

Semester-I

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
						Internal		External		Total		
			Th(L)	Pr	Total	Th	Pr	Th	Pr	Th	Pr	
CA-101	Core	Computer Organization & Architecture	04	-	04	40	-	60	-	100	-	04
CA-102	Core	Database Management System (DBMS)	04	-	04	40	-	60	-	100	-	04
CA-103	Core	Mathematical Foundations of Computer Science-I	04	-	04	40	-	60	-	100	-	04
CA-104	Core	System Programming	04	-	04	40	-	60	-	100	-	04
CA-105	Skill Based	Object Oriented Programming using C++	04	-	04	40	-	60	-	100	-	04
CA LAB-I	Core	LAB on DBMS	-	04	04	-	40	-	60	-	100	04
CA LAB-II	Core	LAB on C++ Programming	-	04	04	-	40	-	60	-	100	04
AC-101	Audit Course	Practicing Cleanliness	-	02	02	-	100	-	-	-	100	02

Semester-II

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
			Th(L)	Pr	Total	Internal		External		Total		
						Th	Pr	Th	Pr	Th	Pr	
CA-201	Core	Automata Theory and Computability	04	-	04	40	-	60	-	100	-	04
CA-202	Core	Data Structures and Algorithms	04	-	04	40	-	60	-	100	-	04
CA-203	Core	Operating systems	04	-	04	40	-	60	-	100	-	04
CA-204	Core	Computer Networks	04	-	04	40	-	60	-	100	-	04
CA-205	Skill Based	Python Programming	04	-	04	40	-	60	-	100	-	04
CA LAB-III	Core	LAB on Data Structures and Algorithms	-	04	04	-	40	-	60		100	04
CA LAB-IV	Core	LAB on Python programming	-	04	04	-	40	-	60		100	04
AC- 201 (A) /(B)/(C)/(D)	Elective Audit Course	Choose one out of four (AC-201 (A)/(B)/(C)/(D)) (Personality and Cultural Development Related)	-	02	02	-	100	-	-	-	100	02

List of Elective Audit Courses to be offered in Semester-II:

AC-201 (A): Soft Skills AC-201 (B): Practicing Sports Activities

AC-201 (C): Practicing Yoga AC-201 (D): Introduction to Indian Music

Note: Syllabus for Spoken Tutorial Courses AC-201 (C)/(D) is available at <https://spoken-tutorial.org> developed at IIT Bombay for MHRD, Government of India.

Semester-III

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
						Internal		External		Total		
			Th(L)	Pr	Total	Th	Pr	Th	Pr	Th	Pr	
CA-301	Core	Web Designing	04	-	04	40	-	60	-	100	-	04
CA-302	Core	Design and Analysis of Algorithms	04	-	04	40	-	60	-	100	-	04
CA-303	Core	Mathematical Foundations of Computer Science-II	04	-	04	40	-	60	-	100	-	04
CA-304	Core	Artificial Intelligence	04	-	04	40	-	60	-	100	-	04
CA-305(A) / (B)	Elective	Choose one from CA-305(A) and CA-305(B)	04	-	04	40	-	60	-	100	-	04
CA LAB-V	Core	LAB on Design and Analysis of Algorithms	-	04	04	-	40	-	60	-	100	04
CA LAB-VI	Core	LAB on Web Designing (HTML/XML/CSS/Navigation)	-	04	04	-	40	-	60	-	100	04
AC-301 (A)/(B)/(C)/(D)	Elective Audit Course	Choose one out of four (AC-301 (A)/(B)/(C)/(D)) (Technology + Value added course)	-	02	02	-	100	-	-	-	100	02

List of Elective Courses to be offered in Semester-III:

CA-305(A): Optimization Algorithms CA-305(B): Data Warehousing and Data Mining (DWDM)

List of Elective Audit Courses:

AC-301 (A) : Computer Skills

AC-301 (B) : Cyber Security

AC-301 (C) : Linux (Spoken Tutorial Course)

AC-301 (D): Advance C++ (Spoken Tutorial Course)

Note: Syllabus for Spoken Tutorial Courses AC-301 (C)/(D) is available at <https://spoken-tutorial.org> developed at IIT Bombay for MHRD, Government of India.

Semester-IV

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
						Internal		External		Total		
			Th(L)	Pr	Total	Th	Pr	Th	Pr	Th	Pr	
CA-401	Core	High Performance Computing Paradigms and applications	04	-	04	40	-	60	-	100	-	04
CA-402	Core	Web Application Technology	04	-	04	40	-	60	-	100	-	04
CA-403	Core	Machine Learning	04	-	04	40	-	60	-	100	-	04
CA-404	Core	Computer Graphics and Digital Image processing	04	-	04	40	-	60	-	100	-	04
CA-405(A) / (B)	Elective	Choose one from CA-405A / CA-405B	04	-	04	40	-	60	-	100	-	04
CA LAB-VII	Core	LAB on Computer Graphics and Digital Image Processing	-	04	04	-	40	-	60		100	04
CA LAB-VIII	Core	LAB on High Performance Computing (HPC) and Web Application Technology	-	04	04	-	40	-	60		100	04
AC-401(A)/(B) / (C)/(D)	Elective Audit Course	Choose one out of four (AC-401(A)/(B)/(C)/(D)) (Professional and Social + Value added course)	-	02	02	-	100	-	-	-	100	02

List of Elective Courses to be offered in Semester-IV:

CA-405A: Accounting and Management Control

CA-405B: Data Analytics

List of Elective Audit Courses:

AC-401(A) : Human Rights

AC-401(B) : Current Affairs

AC-401(C) : Java (Spoken Tutorial Course)

AC-401(D) : PHP & MySQL (Spoken Tutorial Course)

Note: Syllabus for Spoken Tutorial Courses AC-401 (C)/(D) is available at <https://spoken-tutorial.org> developed at IIT Bombay for MHRD, Government of India.

Semester-V

Course Code	Course Type	Title of the Course	Contact Hours/Week			Distribution of Marks for Examination						Credits
			Th(L)	Pr	Total	Internal		External		Total		
						Th	Pr	Th	Pr	Th	Pr	
CA-501	Core	Compiler Construction	04	-	04	40	-	60	-	100	-	04
CA-502	Core	Software Engineering	04	-	04	40	-	60	-	100	-	04
CA-503	Core	Natural Language Processing	04	-	04	40	-	60	-	100	-	04
CA-504	Core	Mobile Application Development	04	-	04	40	-	60	-	100	-	04
CA-505 (A) / (B)	Elective	Choose one from CA-505 (A) or CA-505 (B)	04	-	04	40	-	60	-	100	-	04
CA LAB-IX	Core	LAB on Android Programming	-	04	04	-	40	-	60		100	04
CA LAB-X	Core	LAB on Drupal Framework/LAB on Ruby on Rails	-	04	04	-	40	-	60		100	04

List of Elective Courses to be offered in Semester-V:

CA-505 (A): CMS: Drupal Website Development Framework

CA-505 (B): Ruby on Rails

Semester-VI

Course Code	Course Type	Title of the Course	Contact Hours/Week	Distribution of Marks for Examination					Credits
				Internal		External	Total		
			Th	Pr	Project Evaluation	Th	Pr		
CA-601	Project	Full time Industrial Training	Students contact to teachers through E-mail, AView Software and other ICT technologies throughout the Semester	-	-	Project Evaluation in the form of Demonstration, Project Report Writing, Confidential Report from the Industry, Viva Voce etc.	-	300	12

Semester- I

Course Code:CA-101 **Computer Organization & Architecture**

Clock Hours:60

Total Marks: 100

Course Objectives:

- 1: To enable students to understand significance and various techniques of data (integer numbers, float numbers as well as characters) representation for computers.
- 2: To understand basis building blocks (digital logic circuit and components) Of computers.
- 3: To study the functioning and organization of processor in detail with respect to case study of 8085.
- 4: To study the functioning and organization of memory in detail.
- 5: To elaborate assembly language programing.
- 6: To study how computer interacts with outer world using different I/O modes.

Unit-I

[10] Max Marks:**15**

Number System and Representation of Information: Positional Number System, Conversions, Complements, Representation of Negative numbers, Fixed Point Representation, Floating Point Representation, Character Codes, Error detection Codes

Unit-II

[10] Max Marks:**15**

Digital Logic Circuits and Components: Digital Computers, Boolean Algebra, Boolean Identities, Simplifications of Boolean Expressions, Logic gates, Combinational Circuits: Half-Adder, Full-Adder, Decoders, Encoders, Multiplexers, Sequential Circuits: Flip flops, Registers, Shift Registers, Binary Counters, Memory Unit.

Unit-III

[10] Max Marks:**15**

Processor Organization: General Register Organization, ALU, Instruction codes, Instruction Formats, Stack Organization, Addressing modes, Control Unit, Interrupts

Unit-IV

[10] Max Marks:**15**

8085 Microprocessor: Introduction, History, Internal Architecture, Instruction set, Assembly Language introduction, Simple programs.

Unit-V

[10] Max Marks:**15**

Memory Organization: Types of Memory, Main memory, Auxiliary memory, Memory Hierarchy, Cache Memory and related mapping, replacement policies, Virtual memory.

Unit-VI

[10] Max Marks:15

I/O interface, Asynchronous data transfer, Modes of transfer, Interrupt controlled I/O Transfer, Direct memory access, I/O channels

References:

1. Linda Null and Jullia Lobur (2006). The essentials of Computer Organization and Architecture. Jones Bartlett Publication, ISBN: 978-0763737696
2. M. Morris Mano (2007). Computer System Architecture. Pearson Education India, ISBN: 978-8131700709
3. Ramesh S Goankar(1998). Microprocessor Architecture, Programming & Applications with the 8085. Prentice Hall Publishing (India) Pvt. Ltd.
4. V. Heuring, H. Jorden (2003). Computer Systems Design and Architecture, Pearson Education, Second Ed., 2009. ISBN: 978-0130484406
5. William Stallings (2010). Computer Organization and Architecture Designing for Performance, PHI Pvt. Ltd., ISBN-13: 978-0-13-607373-4.

Course Outcome:

After completion of this course students shall be able to-

- 1: describe number system as well as representation of information in computer.
- 2: Explain simple digital logic circuit and components.
- 3: describe processor and memory organization and functioning in detail.
- 4: solve examples related to memory management.
- 5: discuss I/O process and various modes of I/O
- 6: comprehend various 8085 instructions and categorize as per addressing modes.

Course Code: CA-102

**Database Management System
(DBMS)**

Clock Hours: 60

Total Marks: 100

Course Objectives:

- Understand the basic concepts of database management systems.
- Learn and practice data modelling using the entity-relationship and developing database designs.
- Understand the use of Structured Query Language (SQL) and learn SQL syntax.
- Apply normalization techniques to normalize the database
- Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.

- Understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.
- Understand the functional dependencies and design of the database.
- Understand the concept of Transaction and Query processing
- 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:10

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:15

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:25

Advanced Topics in Databases: Type your unit content here.(comma separated) 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] A. Silberschatz, H.F.Korth, and S.Sudarshan (2011), “Database System Concepts”, TMH Publications, Sixth Edition,2011, ISBN:978-007-132522-6.

Course Outcome:

The learner will be able:

- Describe data models and schemas in DBMS
- Understand the features of database management systems and Relational database.
- Use SQL- the standard language of relational databases.
- Design ER-models to represent simple database application scenarios
- To understand the functional dependencies and design of the database.
- Understand the use of Transaction and Query processing in execution of transaction
- Apply the relational model, specify integrity constraints, and explain how to create a relational database using an ER diagram.
- Analyse and apply two formal query languages, relational calculus and algebra, associated with the relational model.
- Specify how Structured Query Language (SQL) builds upon relational calculus and algebra and effectively apply SQL to create, query and manipulate relational databases.
- Describe how data can be partitioned and distributed across networked nodes of a DBMS, and suggest how queries and segmented data can be optimized and managed in a distributed environment.

Course Code: CA-103

**Mathematical Foundations of
Computer Science-I**

Clock Hours: 60

Total Marks: 100

Course Objectives:

The purpose of the course is to provide the students with several concepts and methods of the number theory, set theory, functions, relations, counting techniques, graph theory and their applications in computer science.

Unit-I

[10] Max Marks:15

Logic and Proofs: Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Proof Methods and Strategy.

Unit-II [10] Max Marks:20

Sets, Relations: Sets, Set Operations, Relations and their Properties, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

Unit-III [10] Max Marks:15

Functions, Function Types, Some Important Functions, Graphical Representation of Functions.

Unit-IV [10] Max Marks:15

Counting: The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations.

Unit-V [10] Max Marks:20

Graphs: Graphs and Graph Models, Graph terminology and Special Types of Graphs, Representing Graphs, nGraph Isomorphism, Connectivity, Euler and Hamilton Paths.

Unit-VI [10] Max Marks:15

Trees: Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

References:

1. Rosen Kenneth H. (2007). Discrete Mathematics and Its Applications, Tata McGraw-Hill Publishing Company Limited, New Delhi, ISBN 13: 978-0-07-064824-1, ISBN 10:0-07-064824-7
2. Liu C. L. (1985). Elements of Discrete Mathematics, Second Edition, Mc Graw Hill, ISBN 007043476X, ISBN 9780070434769

Course Outcome:

Students completing this course will be able to

- 1) Express a logic sentence in terms of predicates, quantifiers, and logical connectives.
- 2) Apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction.
- 3) Use tree and graph algorithms to solve problems.
- 4) Apply the operations of sets and use Venn diagrams to solve applied problems; solve problems using the principle of inclusion-exclusion.
- 5) Determine the domain and range of a discrete or non-discrete function, graph functions, identify one-to-one functions, perform the composition of functions, find and/or graph the inverse of a function, and apply the properties of functions to application problems.

- 6) Solve counting problems by applying elementary counting techniques using the product and sum rules, permutations, combinations.

Course Code: CA-104

System Programming

ClockHours:60

Total Marks:100

Course Objectives:

1. System programming is an essential and important foundation in any computer's application development, and always evolving to accommodate changes in the computer hardware.
2. Distinguish between Systems software and Application software
3. Describe the utility of different system programs & system tools.
4. How an Assembly program or high-level program can be converted to machine instruction.
5. Familiarize with the trade-offs between run-time and compile-time processing (Linking & Loading techniques).
6. Explore the use of compiler with its phases.
7. Use of Syntax directed scheme for intermediate code generation.
8. Construct & use of different compiler tools for code generation & optimization.

Unit-I

[06] Max Marks:12

Introduction to System Programming: Introduction, Programming Languages and Language Processors, Language Processing Activities, Program Execution, Multi-Pass Organization of Language Processors, Symbol Table Entry Formats.

Unit-II

[12] Max Marks:20

Assemblers: Elements of Assembly Language Programming, Simple Assembly Scheme, Pass Structure of Assemblers, Design of a Two-Pass Assembler-Pass I of the Assembler Intermediate Code Forms, Intermediate Code for Imperative Statements, Processing of Declarations and Assembler Directives, Pass II of the Assembler.

Unit-III

[10] Max Marks:12

Macros and Macro Preprocessor: Macro Definition and Call, Macro Expansion, Nested Macro Calls, Conditional Expansion, Expansion Time Loops, Semantic Expansion, Design of a Macro Preprocessor- Design Overview.

Unit-IV

[10] Max Marks:15

Linkers and Loaders: Introduction, Relocation and Linking Concepts, Design of a Linker, Self-Relocating Programs, Linking of Overlay Structured Programs, Dynamic Linking, Loaders.

Unit-V [06] Max Marks:10

Scanning and Parsing: Programming Language Grammars, Scanning, Parsing – Top Down Parsing and Bottom Up Parsing.

Unit-VI [10] Max Marks:20

Compilers: Causes of a Large Semantic Gap, Binding and Binding Times, Static and Dynamic Memory Allocation, Compilation of Expressions – Intermediate Codes for Expressions, Postfix Notation, Triples and Quadruples, Function and Procedure Calls, Code Optimization- Optimizing Transformations, Local Optimization and Introduction to Global Optimization, Pure and Impure Interpreter.

Unit-VII [06] Max Marks:11

Software Tools: Software Tools for Program Development, Editors, Debug Monitors.

References:

1. Dhamdhere D M, (2011). System Programming. Tata McGraw Hill Education Private Limited, New Delhi. ISBN 10: 0071333118
2. Leland L. Beck, D. Manjula (2002). System Software: An Introduction to System Programming, 3rd Edition, Pearson Education, ISBN-10: 817758555X
3. John J. Donovan (2001). System Programming. Indian Edition, Mc Graw Hill Education, ISBN10:0074604821
4. Srimanta Pal, (2011). System Programming. Oxford Higher Education, ISBN-10: 0198070888

Course Outcome:

1. System programming leads to the development of computer system software that manages and controls the computer operations
2. The system programming enhances or extends the functions of an operating system by developing components such as drivers, utilities and updates.
3. Apply the knowledge & technique to develop solutions to real world problems by compiling application programs.
4. Ability to identify, formulate, and solve computer engineering problems with proper systematic & semantic approach.
5. Develop possible program constructs for further code generation with Type checking & memory management strategy.
6. Design a simple compiler with tools to optimized results.

Course Code: CA-105

**Object Oriented Programming
using C++**

Clock Hours: 60

Total Marks: 100

Course Objectives:

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

Unit-I

[10] Max Marks:12

Fundamentals

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

Object-Oriented Programming using C++ : C++ programming Basics, Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, class scope and accessing class members, separating interface from implementation, controlling access to members.

Functions: Function, function prototype, accessing function and utility function, Constructors and destructors, Copy Constructor, Objects and Memory requirements, Static Class members, data abstraction and information hiding, inline function

Unit-II

[12] Max Marks:20

Polymorphism and Inheritance

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

Inheritance- Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Inheritance, Public and Private Inheritance, Levels of Inheritance, Multiple Inheritance, Ambiguity in Multiple Inheritance, Aggregation, Classes Within Classes.

Polymorphism: concept, relationship among objects in inheritance hierarchy, abstract classes, polymorphism.

Unit-III

[12] Max Marks:18

Virtual Functions: Pointers- indirection Operators, Memory Management: new and delete, Pointers to Objects, A Linked List Example, accessing Arrays using pointers, Function pointers, Pointers to Pointers, A Parsing Example, Debugging Pointers, Dynamic Pointers, smart pointers, shared pointers, Case Study : Design of Horse Race Simulation. Virtual

Function Friend Functions, Static Functions, Assignment and Copy Initialization, this Pointer, virtual function, dynamic binding, Virtual destructor.

Unit-IV [10] Max Marks:18

Templates: function templates, Function overloading, overloading Function templates, class templates, class template and Nontype parameters, template and inheritance, template and friends Generic Functions, Applying Generic Function, Generic Classes, The typename and export keywords, The Power of Templates.

Exception Handling: Fundamentals, other error handling techniques, simple exception handling Divide by Zero, rethrowing an exception, exception specifications, processing unexpected exceptions, stack unwinding, constructor, destructor and exception handling, exception and inheritance.

Unit-V [08] Max Marks:16

Files and Streams: Data hierarchy, Stream and files, 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, Early vs. Late Binding.

Unit-VI [08] Max Marks:16

Standard Template Library (STL): Introduction to STL, Containers, algorithms and iterators, *Containers:* Sequence container and associative containers, container adapters, *Algorithms:* basic searching and sorting algorithms, min-max algorithm, set operations, *Iterators:* input, output, forward, bidirectional and random access. Object Oriented Programming – a road map to future

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. Deitel, C++ How to Program, 7th Edition, Pearson Education, ISBN-10: 0-13-611726-0 ISBN-13: 978-0-13-611726-1.
4. Herbert Schildt, C++ The complete reference, Fifth Edition, McGraw HillProfessional, , ISBN-10: 0071634800 ISBN-13: 978-0071634809.
5. Stanley B. Lippman, Josée Lajoie, Barbara E. Moo(2013), C++ Primer, Fifth Edition, Addison-Wesley, 2013, ISBN-13: 978-0-321-71411-4, ISBN-10: 0-321-71411-3

Course Outcome:

By the end of this course, students should be able to:

- Understand and use the basic programming constructs of C++
- Manipulate various C++ datatypes, such as arrays, strings, and pointers
- Use memory appropriately, including proper allocation/deallocation procedures
- Write small-scale C++ programs using the above skills

Course Code: CA LAB-I

LAB on DBMS

Total Marks: 100

Course Objectives:

- Provides foundation knowledge in database concepts, technology and practice to prepare students into expert database application developers.
 - Strong practice in SQL programming through a variety of database problems.
 - Develop database applications using front-end tools and back-end DBMS.
15. Creating database tables and using data types.
 - Create table
 - Modify table
 - Drop table
 2. Practical Based on Data Manipulation.
 - Adding data with Insert
 - Modify data with Update
 - Deleting records with Delete
 3. Practical Based on Implementing the Constraints.
 - NULL and NOT NULL
 - Primary Key Constraint
 - Foreign Key Constraint
 - Unique Constraint
 - Check Constraint
 - Default Constraint
 15. Practical for Retrieving Data Using following clauses.
 - Simple select clause
 - Accessing specific data with Where
 - Ordered By
 - Distinct
 - Group By
 15. 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, D2K, 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:

- Design and implement a database schema for a given problem-domain
- Create and maintain tables using PL/SQL
- Populate and query a database using SQL DML/DDDL commands.
- Programming PL/SQL including stored procedures, stored functions, cursors, triggers.
- Prepare reports and forms
- Application development using PL/SQL & front end tools

Course Code: CA LAB-II

LAB on C++ Programming

Total Marks: 100

Course Objectives:

- Apply object-oriented approaches to software problems in C++
 - Apply exception handling techniques to software problems in C++
 - Apply generic programming approaches using templates
 - Efficiently use standard template library in software development
1. Write program to demonstrate encapsulation using of class.
 2. Write program to demonstrate use of constructor, constructor overloading and destructor.
 3. Write program to demonstrate use of array manipulations.
 4. Write program to demonstrate use of string manipulations.
 5. Write program to demonstrate use of function overloading.
 6. Write program to demonstrate use of operator overloading.
 7. Write program to demonstrate use of pointers.
 8. Write program to demonstrate use of operator overloading.
 9. Write program(s) to demonstrate use of all types of inheritance.
 10. Write program to demonstrate use of virtual function.
 11. Write program to demonstrate use of friend function.
 12. Write program to demonstrate use of virtual class.

13. Write program to demonstrate use of function templates.
14. Write program to demonstrate use of class templates.
15. Write program to demonstrate use of static data member and static member function.
16. Write program to demonstrate use of exception handling.
17. Write program to demonstrate command line arguments.
18. Write program to demonstrate file handling.
19. Write program(s) to demonstrate use of STL.

Course Outcome:

By the end of this course, students should be able to:

- Understand the object-oriented approach for the program development
- Develop logic of a program for solving real time problems.
- Isolate and fix common errors in C++ programs
- Design problem solutions using Object Oriented techniques.
- Apply the concepts of data abstraction, encapsulation, polymorphism, overloading, and inheritance for problem solutions.
- Use the OOP concepts of C++ appropriately in problem solving.

Semester- II

Course Code: CA-201

**Automata Theory and
Computability**

Clock Hours: 60

Total Marks: 100

Course Objectives:

To cover all aspects of theoretical computer science namely automata, formal languages, computability, and basics of complexity

Unit-I

[12] Max Marks:20

Finite Automata:

Sets, relations, functions, graphs, trees, mathematical induction, Finite Automata(FA), definition, description, transition systems, acceptability of a string, NFA, DFA, equivalence of DFA and NFA, Melay & Moore model, minimization of automaton, Applications.

Unit-II

[04] Max Marks:12

Formal Languages:

Formal languages, Chomsky classification of languages, languages, their relation and automaton.

Unit-III

[14] Max Marks:23

Regular Expressions:

Regular expressions, FA and regular expressions, pumping lemma for regular sets, applications of pumping lemma, closure properties of regular sets, regular sets and regular grammars.

Unit-IV

[10] Max Marks:15

Context Free Languages:

CFLs and derivation trees, ambiguity in Context-Free Grammars (CFGs), simplification of CFGs, Normal Forms for CFGs (CNF and GNF), pumping lemma for CFLs, decision algorithms for CFLs.

Unit-V

[08] Max Marks:15

Push Down Automata:

Pushdown Automaton (PDA), informal description, basic definitions, acceptance by a PDA, PDA and CFLs.

Unit-VI

[08] Max Marks:15

Turing Machine:

Turing Machine, Model, computable languages and function, representation of TMs, Language Acceptability by TMs, Design of TM, Halting Problem of TMs.

References:

1. Smita Rajpal. Theory of Automata and Formal Languages: GALGOTIA Publications ISBN 1234027054.
2. J.E.Hopcraft, R. Motwani and J.D.Ullman. Introduction to Automata Theory languages & Computation: Pearson Education Asia ISBN 978-0321455369.
3. K.L.P.Mishra, N. Chandrashekharan. Theory of Computer Science: PHI ISBN 8120329686.
4. Martin John C. Introduction to Language & Theory of computation: TMH ISBN 9780070660489.
5. M.Sipser (1997). Introduction to the Theory of Computation; Singapore: Brooks/Cole, Thomson Learning ISBN 978-1133187790.
6. <http://nptel.ac.in/>

Course Outcome:

Learn several formal mathematical models of computation along with their relationship with formal languages. Learn regular languages and context free languages which are crucial to understand how compilers and programming languages are built. Understanding about limitations on what computers can do and learn that certain problems are unsolvable.

Course Code: CA-202

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
- 3 To impart knowledge of advance topics like AVL Trees, BTrees, B* and B+ Trees
- 4 To understand concepts about searching and sorting techniques
- 5 Apply hashing concepts for a given problem
- 6 To understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures

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, B-Trees: insertion and deletion, Introduction to B+ and B* Trees

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:

1. Student will be able to choose appropriate data structure as applied to specified problem definition.
2. Students will be able to use linear and non-linear data structures like stacks, queues, linked list etc.
3. To increase the student's intuitive understanding of search trees
4. To learn advanced tree data structures
5. To learn to represent data using graph data structure
6. Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
7. Students will be able to apply concepts learned in various domains like DBMS, compiler construction etc.

Course Code: CA-203

Operating Systems

Clock Hours: 60

Total Marks: 100

Course Objectives:

Students will try to learn:

1. To understand the main components of an OS & their functions with the concept of system calls.
2. To study the evaluation of OS, computer organization interface
3. To understand the various issues in process management, scheduling and concept of threads.
4. To study the scheduling algorithms, the concept of process synchronization and deadlocks.
5. 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
6. To study the need for special purpose operating system with the advent of new emerging technologies

Unit-I

[04] Max Marks:08

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:06

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:06

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. 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:

Students will able to:

1. Describe the important computer system resources and the role of operating system in the management policies and algorithms.
2. Understand the process management policies and scheduling of processes by CPU.
3. Evaluate the requirement for process synchronization and coordination handled by operating system
4. Describe and analyse the memory management and its allocation policies.
5. Identify use and evaluate the storage management policies with respect to different storage management technologies.
6. Identify the need to create the special purpose operating system.

Course Code:CA-204

Computer Networks

ClockHours:60

Total Marks:100

Course Objectives:

1. This course introduces students to local, metropolitan, and wide area computer network and focusses on building a firm foundation for understanding Data Communications and Computer Networks.
2. It is based around the OSI Reference Model, that deals with the major issues in the bottom three (Physical, Data Link and Network) layers of the model.
3. Also, student learn how data transmission take place between nodes and what are the difficulties encounter during transmission.
4. Introduction to the Internet protocol suite and its development.
5. Students are also introduced to the areas of Cryptography and Network Security.
6. To expose students to emerging technologies and their potential importance.
7. This course provides the student with fundamental knowledge of the various aspects of computer networking and enables students to appreciate recent developments in this area.

Unit-I

[10] Max Marks:20

Overview of Computer Network, OSI and TCP/IP Reference Models, Guided and Unguided Transmission Media, Analog and Digital Communication, Encoding and Modulation, Nyquist Theorem, Shannon's capacity, Switching techniques- TDM,FDM.

Unit-II

[15] Max Marks:25

Framing, Error detection and Error correction- Vertical Redundancy Check (VRC) Longitudinal Redundancy Check (LRC), Hamming Distance, Hamming Code, Cyclic Redundancy Check(CRC), Stop and Wait Protocol, Sliding Window Protocol, Go-Back-n ARQ, Selective-Reject ARQ, HDLC.

Unit-III

[10] Max Marks:15

Channel Allocation, ALOHA Systems, CSMA Protocols, Collision Free Protocols, Local Area Networks, Bridges, ATM

Unit-IV [15] Max Marks:25

Routing- Flooding, Shortest Path Routing, Distance Vector Routing, Link State Routing, Congestion Control-Leaky Bucket and Token Bucket algorithms, IP Protocol, IP Addressing, ARP, RARP, OSPF, BGP, TCP, UDP,DHCP.

Unit-V [10] Max Marks:15

Cryptography and Network Security- Introduction to the Concept of Security, Cryptographic Techniques, Computer-based Symmetric and Asymmetric Key Cryptographic Algorithms, Public Key Infrastructure (PKI), Internet Security Protocols, Network Security.

References:

- 1.Andrew S. Tanenbaum ,(2009). Computer Networks 4th Edition , Pearson Prentice Hall.ISBN- 978-81-7758-165-2
- 2.Behrouz A. Forouzan, (2012).Data Communications and Networking. 5th Edition. Mc Graw – Hill, ISBN-10: 0073376221
3. William Stalling, (2013) Cryptography and Network Security 6 edition Pearson Education India, ISBN-10: 9332518777
4. Behrouz A Forouzan and Debdeep Mukhopadyay, (2010) Cryptography and Network Security, 2nd edition, McGraw Hill Education, ISBN-10: 007070208X
5. Atul Kahate, (2009),Cryptography and Network Security, 2nd Edition, McGraw Hill Education, ISBN-10: 0070151458

Auxiliary Resources:

- a. <http://nptel.ac.in/courses/106105081/1>
- b. <http://www.nptelvideos.in/2012/11/computer-networks.html>
- c. <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-033-computer-system-engineering-spring-2009/video-lectures/lecture-9/>

Course Outcome:

1. Have a good understanding of the OSI Reference Model and have an upright knowledge of Layers 1-3.
2. To be familiar with contemporary issues in networking technologies
3. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies;
4. Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols;
5. Apply the knowledge of cryptographic utilities and authentication mechanisms to design secure applications

6. Have good exposure to cryptography and network security and in future may think of implementing such techniques.

Course Code: CA-205

Python Programming

Clock Hours: 60

Total Marks: 100

Course Objectives:

- To learn easy-to-read, easy-to-maintain, interactive, portable programming language with a broad standard library.
- To introduce object-oriented programming using simple and an easy-to-use language.
- To learn regular expressions, exception handling, file handling techniques and GUI constructs for web
- To learn modular and package oriented programming
- To introduce programming for natural language processing and machine learning.

Unit-I

[12] Max Marks:20

The Python Programming Language, Python Data, Variables, Expressions and Statements, Values and Data Types, Type conversion Functions, Operators and Operands, Input, Order of Operations, Functions, Calling Functions, Passing Functions, Formal Arguments, Variable-length Arguments, Functional Programming, Boolean Expressions, Logical operators, Precedence of Operators, Conditional Execution, Unary Selection, Nested conditionals, Chained conditionals, Boolean Functions, Iteration, The for loop, The while Statement

Unit-II

[08] Max Marks:15

Strings, A Collection Data Type, Operations on Strings, Index Operator: Working with the Characters of a String, String Methods, Length, The Slice Operator, String Comparison, Lists, List Values, List Length, Accessing Elements, List Membership, Concatenation and Repetition, List Slices, Lists are Mutable, List Deletion, Objects and References, Aliasing, Cloning Lists, Repetition and References, List Methods, Append versus Concatenate Lists and for loops, Using Lists as Parameters, Nested Lists, Strings and Lists, List Type Conversion Function, Tuples, Tuple operators and built-in functions, Tuples and Mutability, Tuple Assignment, Tuples as Return Values

Unit-III

[15] Max Marks:20

Dictionaries, Dictionary Operations, Dictionary Methods, Dictionary Keys, Aliasing and Copying, Sparse Matrices, Working with Data Files, Finding a File on your Disk, Reading a File, Iterating over lines in a file, Writing Text Files, Object Oriented Programming, Classes, Instances, Class method Calls, Coding Class Tree, Attributes, Building and Method Invocation, Composition, Inheritance, Operator Overloading, Encapsulation and Information Hiding, Search Algorithms, Sorting Algorithms, Hash Tables

Unit-IV

[10] Max Marks:20

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

[15] Max Marks:25

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

References:

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

Course Outcome: Upon completion of the course, the students will be able to

- use lists, tuples, dictionaries and strings efficiently for solving real world problems.
- discuss the concepts of object-oriented programming.
- use files to read and write data easily and efficiently.
- develop GUI based programming for web.
- develop exception handling code
- develop modules and packages.

Course Code: CA Lab-III LAB on Data Structures and Algorithms Total Marks: 100

Course Objectives:

The main objectives of this course are:

- Solve real-world problems by reasoning about data structure choices, choose appropriate implementations.
- Identify the strengths and weaknesses of different data structures
- To make the students write various programs and ADTS for all data structures.
- Students will learn to write, debug, and test large programs systematically.
- Think critically for improvement in solutions.
- Determine which algorithm or data structure to use in different scenarios.

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
- Balanced Binary Tree (AVL)
- B-Trees

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:

The above exercise shall make the students competent in the following ways and will be able to learn following parameters at the end of the course.

- Develop solutions for a range of problems using procedure oriented / object oriented programming.
- Apply divide and conquer strategy to searching and sorting problems using iterative And / or recursive solutions.
- Use critical thinking skills and creativity to solve the problems.
- Design scenarios to explain behaviors and demonstrate correctness of programs.
- Determine which algorithm or data structure to use in different scenarios.
- Choose the appropriate data structure and algorithm design method for a specified application.
- Have practical knowledge on the applications of data structures

Course Code: CA Lab-IV Lab on Python Programming Total Marks: 100

Course Objectives:

- To acquire programming skills in core Python.
 - To acquire Object Oriented Skills in Python.
 - To develop the skill of designing Graphical user Interfaces in Python
 - 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. Develop programs to learn GUI programming using Tkinter.
 8. 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
 9. Demonstrate the concept of String-Based Exceptions, Class-Based Exceptions and Nesting Exception handlers.
 10. Demonstrate implementation of the Anonymous Function Lambda.
 11. Demonstrate implementation Mapping Functions over Sequences.

12. Demonstrate implementation functional programming tools such as filter and reduce
13. Demonstrate the Module Creation, Module usage, Module Namespaces, Reloading Modules, Module Packages, Data Hiding in Modules.
14. Demonstrate Mixed Usage Modes of modules, Changing the Module Search Path, The import as Extension, Relative Import Syntax, Module Design Concepts

Course Outcome:

At the end of the course, the student will be able to

- implement basic constructs of Python programming language
- implement object oriented concepts
- implement GUI applications
- implement file handling applications
- implement exception handling applications
- implement modular programs

Semester- III

Course Code: CA-301

Web Designing

Clock Hours: 60

Total Marks: 100

Course Objectives:

Develop skills in analyzing the usability of a web site. Understand how to plan and conduct user research related to web usability. Learn the language of the web: HTML, XML and CSS. Learn techniques of responsive web design, including media and queries.

Unit-I [05] Max Marks:10

Introduction to Web Development: Components of Web application, Static vs. dynamic web pages, A survey of browsers, servers and scripting languages, URL, Introduction to HTML, HTML5, XHTML and CSS, Semantic markup, Web Page Designing Principles, Page layout, Website organization

Unit-II [10] Max Marks:15

HTML 5.0: Structure of HTML5 Document, Elements, tags and attributes, DOCTYPE declaration, HEAD section, Coding text elements, Headings and paragraphs, Inline elements for formatting and emphasizing text, div and span elements, Events, WEB Form2.0, SVG, Canvas, Audio & Video, Geolocation, Drag & Drop, Web Workers.

Unit-III [10] Max Marks:20

CSS to format elements of a web page: CSS applications, Selectors: Basic Rule, Grouping, class and ID selectors, attribute selectors, document structure, Specificity, Inheritance, cascade, Values and Units, Fonts, Colors Working with text, text alignment, spacing(letter/word), Text decoration, Text Shadows, CSS box model, Spacing, borders and backgrounds, Page layout using CSS, Floating elements, Multicolumn layout, Ways to include CSS in a web page.

Unit-IV [15] Max Marks:20

HTML and CSS skills: Working with links and lists, Working with images, Working with tables, Working with forms, Adding audio, video to web pages, JavaScript for image rollovers, image swaps, slide shows, tabbed data and data validation

Unit-V [10] Max Marks:20

Core Concepts of XML: Parts of an XML document, Document Prolog, The XML declaration, The document type declaration, Elements, Attributes, Namespaces, Entities, Comments, CDATA Section, Processing instructions, XSL, Displaying XMLwith CSS, JAVA Script, XSLT

Unit-VI

[10] Max Marks:15

Modeling Information: Simple Data storage, Dictionaries, Records, XML and Databases, Narrative Documents, Flows and sections, Blocks and inlines, Complex structures, Metadata, Linked Objects

References:

1. Anne Boehm, Murach's HTML, XHTML and CSS, Shroff Publishers and Distributers Pvt. Ltd., ISBN 13: 978-93-5023-095-4
2. Learning XML, Erik T. Ray, O'Reilly, Shroff Publishers and Distributers Pvt. Ltd., ISBN 13: 978-81-8404-896-4
3. Learning Web Design
4. Eric A. Meyer, CSS: The Definitive Guide, Visual Presentation for the Web, 3rd Edition, O'Reilly Media, November 2006, ISBN:978-0-596-52733-4| ISBN 10:0-596-52733-0(Print), ISBN 978-0-596-15940-5, 10:0-596-15940-4(Ebook)
5. Benoit Marchal, XML by Example, 2nd Edition, Microsoft Press 2001, ISBN-10: 0-7897-2504-5, ISBN-13: 978-0-7897-2504-2.

Auxiliary Resources:

<https://www.tutorialspoint.com/html5/>

Course Outcome:

Understand how web works really, what makes web sites work.

Simple and impressive web page design techniques from basics to advanced to focus on goal oriented and user centric designs.

Setting up page layouts, color schemes, typography in the designs.

Writing valid and concise code for webpages using XML, CSS and HTML.

Course Code: CA-302

Design and Analysis of Algorithms

Clock Hours: 60

Total Marks: 100

Course Objectives:

To Understand and learn

1. Basic concepts of algorithms and analyze the performance of algorithms.
2. Algorithm design techniques for developing algorithms.
3. Searching and traversal algorithms for graphs.
4. Nondeterministic algorithms and NP class of problem.

Unit-I

[10] Max Marks:15

What Is An Algorithm?, Algorithm Specification, Pseudocode Conventions, Recursive Algorithms, Complexity, Asymptotic Notation, Practical Complexities And Performance Measurement

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

Unit-II [12] Max Marks:15

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

Unit-III [08] Max Marks:15

Greedy Method: General Method, Knapsack Problem, Huffman Code, Minimum-Cost Spanning Trees (Prim's & Kruskal's Algorithm), Optimal Storage On Tapes, Single-Source Shortest Paths.

Unit-IV [08] Max Marks:15

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

Unit-V [06] Max Marks:15

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

Unit-VI [08] Max Marks:15

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

Unit-VII [08] Max Marks:10

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:

1. Horowitz E. and Sahni S. "Fundamentals of computer Algorithms" Galgotia publications.
2. Horowitz E., Sahni S. and Rajshekaran S(), Computer Algorithms, Computer Science Press.
3. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani(2006), Algorithms. McGraw-Hill publications. ISBN 9780073523408
4. Cormen, Leiserson and Rivest, Introduction to Algorithms,; Prentice Hall of India ISBN: 978-81-203-4007-7

Course Outcome:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Design and analyze divide-and-conquer based algorithms.

4. Devise and Synthesize greedy and dynamic-programming based algorithms.
5. Employ graphs to model problems solvable using traversal techniques.
6. Able to model problems using backtracking
7. Able to classify nondeterministic polynomial time algorithms.

Course Code: CA-303

**Mathematical Foundations of
Computer Science-II**

Clock Hours: 60

Total Marks: 100

Course Objectives:

1. To build the foundation of computer algorithms using mathematical base.
2. To introduce the concepts of induction to prove certain property is true for non-negative integers.
3. To introduce the concepts of recursive data types, linear recurrences, divide and conquer recurrences and solving recurrences.
4. To introduce the concept of generating functions and applications of principle of inclusion and exclusion.
5. To apply statistical measures on the data and represent it graphically.
6. To relate practical examples to the probability theory and probability distributions to build the foundation for machine learning.
7. To understand stochastic processes and apply Markov chain theory to relate real time problems.
8. To understand hidden markov model and Chapman-Kolmogorov equation for solving problems.

Unit-I

[15] Max Marks:25

Induction and Recursion: Mathematical Induction, Strong Induction and Well Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness, The Towers of Hanoi, Merge Sort, Linear Recurrences, Divide-and-Conquer Recurrences, A Feel for Recurrences

Unit-II

[15] Max Marks:20

Advance Counting Techniques: Recursive Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

Unit-III

[05] Max Marks:10

Statistics: Population and sample, parameters and statistics: definition, types: Descriptive and Inferential, applications, Descriptive Statistics: Mean, median, mode and standard deviation, variance, Graphical statistics

Unit-IV

[15] Max Marks:20

Probability: Making decisions under uncertainty, Classical definition of Probability, Events and their Outcomes, Rules of Probability, Probability axioms, Random variables (discrete and continuous), Joint and Conditional probability, independence and Bayes theorem, Distribution of a random vector, Probability mass function, Probability density function and distribution function. Distributions: Binomial, Poisson, Uniform, Exponential, Normal.

Unit-V

[10] Max Marks:15

Stochastic Processes: Definitions and classifications of Stochastic Processes, discrete and continuous Markov models, Hidden 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. Bhat B.R, Srivenkatramana T and Rao Madhava K.S. (1996): Statistics: A Beginner's Text, Vol. I, New Age International (P) Ltd.
4. Croxton F.E, Cowden D.J and Kelin S (1973): Applied General Statistics, Prentice Hall of India ISBN 10: 0201089947 ISBN 13: 9780201089943
5. Hogg, Robert V. & Craig Allen T. (2008). Introduction to Mathematical Statistics, Pearson Education ISBN 0-02-978990-7
6. Goon A.M., Gupta M.K., Dasgupta. B. (2001), Fundamentals of Statistics, Volume I and II, World Press, Calcutta.
7. Ross, S. (2005). Introduction to Probability Models, (6th Ed. Academic Press). ISBN 978-0-12-375686-2
8. Medhi, J. (1994). Stochastic Processes, (2nd Ed. New Age Publisher) ISBN : 978-93-86286-48-2

Course Outcome:

Students are expected to know and be able to

1. demonstrate their understanding of and apply methods of mathematics in computer science to subsequent courses in algorithm design and analysis.
2. identify, formulate, and develop solutions to computational challenges.
3. understand and solve a computational problem to meet desired needs within realistic constraints.
4. analyze the behavior of the data, model the data using statistical measures and represent it graphically on paper without using available computerized tools.
5. 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-304

Artificial Intelligence

Clock Hours: 60

Total Marks: 100

Course Objectives:

The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. Emphasis will be placed on the teaching of these fundamentals, not on providing a mastery of specific software or tools programming environments.

The student should be made to:

- 1) Gain a historical perspective of AI and its foundations.
- 2) Study the concepts of Artificial Intelligence.
- 3) Investigate applications of AI techniques in intelligent agents
- 4) Learn the methods of solving problems using Artificial Intelligence.
- 5) Learn various peculiar search strategies for AI

Unit-I [08] Max Marks:10

Introduction: Overview and Historical Perspective, Turing test, Physical Symbol Systems and the scope of Symbolic AI, Agents.

Unit-II [06] Max Marks:10

State Space Search: Depth First Search, Breadth First Search, DFID.

Unit-III [08] Max Marks:12

Heuristic 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

Planning and Constraint Satisfaction: Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graphplan, Constraint Propagation.

Unit-VIII [08] Max Marks:15

Logic and Inferences: Propositional Logic, First Order Logic, Soundness and Completeness, Forward and Backward chaining.

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:

At the end of the course, the student should 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.
- 4) Apply the suitable algorithms to solve AI problems

Course Code:CA-305 (A)

Optimization Algorithms

Clock Hours:60

Total Marks: 90

Course Objectives:

1. To introduce with the branch of OR and its role in decision making.
2. To list out various types of applications of operations research (OR).
3. To explain Linear Programming Problem (LPP) and practice with techniques to solve various types of LPP (transportation problem, assignment problems, special cases of duality, Integer programming problems)
4. Describe the significance, concept of game theory and algorithms to solve game theory problems.
5. Introduce critical path analysis using network problems.

Unit-I

[05] Max Marks:08

Overview of operations Research: Introduction, Applications, Role of OR in Decision Making, Feasible and optimal Solutions

Unit-II

[15] Max Marks:20

Linear Programming: Special Types: Transportation Problem as LPP, Initial Basic Feasible Solution, North West corner Rule, Lowest Cost Method, Vogel's Approximation Method, MoDi method for optimization, Degeneracy.

Assignment problem, Hungarian Method, Special cases of assignment problem

Unit-III [18] Max Marks:24

Linear Programming Problems: Introduction, Formulation of Mathematical model of LPP, Standard form of linear programming problems, Solving LPP using Graphical method, Infeasible LPP, Unbounded LPP, Basic feasible solutions, Simplex method for solving LPP, augmentation using Slack and artificial variables, Big M and two phase method, Degeneracy, alternative optima, Interpretation of final Simplex table, Duality: concept, applications and example.

Unit-IV [06] Max Marks:08

Integer Programming: Introduction, How it differs from LPP, Pure and mixed integer programming problems, Binary IPP, Techniques to solve IPP.

Unit-V [08] Max Marks:15

Network Models: Definitions, Applications, Representation of a problem in network form, Critical Path Analysis, Resource planning, Giantt Chart.

Unit-VI [08] Max Marks:15

Game Theory : Concept, Two party zero sum game, Pay off matrix, Pure and mixed strategy games, Rule of Dominance, Subgame method, Brown's Algorithm

References:

1. Hamdy Taha (2010). Operations Research: An Introduction. Pearson Education. ISBN: 978-0132555937
2. L C Jhamb. Quantitative Techniques For Managerial Decisions Vol I, Vol II. Everest Publishing House, ISBN: 8186314628
3. PanneerSelvan R (2006). Operations Research. Prentice Hall of India. ISBN: 978-8120329287

Course Outcome:

After completion of this course students shall be able to-

1. write about OR and decision making.
2. Differentiate between feasible and optimal solution
3. Apply solving techniques to all types of LPP.
4. Apply solving techniques to network problems and game theory problems as well.

Course Code: CA-305 (B)

Data Warehousing and Data Mining (DWDM)

Clock Hours: 60

Total Marks: 100

Course Objectives:

1. To comprehend evolution of decision making, operational vs decision support system and the concept of data warehouse.
2. To understand transactional and analytical processing
3. Significance of analytical processing and importance of data pre-processing.
4. Learn various data pre-processing techniques, methods.
5. Understand and apply various techniques/algorithms to obtain meaningful patterns from data (Association mining, classification and clustering)

Unit-I

[08] Max Marks:12

Introduction to Data Warehousing: Evolution of decision system, Failure of past decision support system, Operational v/s decision support systems, Data warehousing lifecycle, Architecture, Building blocks, Components of DW, Data Marts and Metadata

Unit-II

[08] Max Marks:12

Data Pre-processing: need for pre-processing of the data, Descriptive data summarization, Data cleaning, Data Integration and transformation, Data reduction, Data discretization and concept hierarchy generation.

Unit-III

[10] Max Marks:15

OLAP Analytical Processing: OLAP in Data warehouse, Demand for online analytical processing, need for multidimensional analysis, limitations of other analysis methods, OLAP definitions and rules, OLAP characteristics, major features and functions, OLAP models-ROLAP, MOLAP, HOLAP, Differentiation, Data cubes and operations on cubes.

Unit-IV

[06] Max Marks:09

Data Mining: Introduction-Data Mining functionalities, Classification of Data Mining Systems, basic Data Mining task, Data Mining Issues

Unit-V

[08] Max Marks:12

Association Rule Mining: Efficient and Scalable Frequent Item set Mining Methods – Mining Various Kinds of Association Rules – Association Mining to Correlation Analysis – Constraint-Based Association Mining.

Unit-VI

[10] Max Marks:15

Classification and Prediction :Issues Regarding Classification and Prediction – Classification by Decision Tree Introduction – Bayesian Classification – Rule Based Classification –

Classification by Back propagation – Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods – Prediction – Accuracy and Error Measures – Evaluating the Accuracy of a Classifier or Predictor – Ensemble Methods – Model Section.

Unit-VII

[10] Max Marks:15

Cluster Analysis :Types of Data in Cluster Analysis – A Categorization of Major Clustering Methods – Partitioning Methods – Hierarchical methods – Density-Based Methods – Grid-Based Methods – Model-Based Clustering Methods – Clustering High-Dimensional Data – Constraint-Based Cluster Analysis – Outlier Analysis

References:

1. Jiawei Han and MichelineKamber “Data Mining Concepts and Techniques” Second Edition, Elsevier, Reprinted 2008.
2. M. H. Dunham. Data Mining: Introductory and Advanced Topics. Pearson Education. 2001.
3. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann. 2000.
4. D. Hand, H. Mannila and P. Smyth. Principles of Data Mining. Prentice-Hall. 2001
5. Tan Steinbach, Vipin Kumar, Introduction to Data mining, Pearson Education
6. Jarke Vassiliou, Fundamentals of Data Warehouses, IInd Edition, Springer
7. Anahory Murray, Data Warehousing in Real World, Pearson Education
8. Paulraj Ponniah , Data Warehousing.

Course Outcomes:

After this course students shall be able to –

1. Explain organization of data warehousing and data marts.
2. Differentiate between OLTAP and OLAP
3. Apply data pre-processing techniques
4. Write basic algorithms for extracting patterns from data (association mining, classification and clustering)
5. Solve problems related with various aspects of data mining.

Course Code: CA Lab-V

LAB on Design and Analysis of Algorithms

Total Marks: 100

Course Objectives:

Understand and learn

1. To convert the algorithms to code.
2. To measure the complexities at run time.
3. To modify the algorithms for efficiency.
4. To debug and test the programs.
5. To conclude using profile of outcomes.

Laboratory Requirements:

OS: Windows/Linux,

Programming Language: C++/Java/C#

1. Write a program for creating max./min. heap using
 - INSERT
 - ADJUST/HEAPIFY
2. Write a program to implement union and find operation.
3. Write a program to find minimum and maximum form a given array.
4. Write a program for searching element form given array using binary search for n=1000,2000,3000 find exact time of execution.
5. 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
6. Write a program for matrix multiplication using Strassen's matrix multiplication.
7. Write a program to find solution of Knapsack instant.
8. Write a program to find shortest path using single source shortest path.
9. Write a program to find Minimum-Cost Spanning Trees (Prim's & Kruskal's Algorithm).
10. Write a program to find shortest path using all pair path.
11. Write a program to find longest common subsequence.
12. Write a program to implement breadth first and depth first search.
13. Write a program to implement breadth first and depth first traversal.
Write a program to find all solutions for 8-queen problem using backtracking.

Course Outcome:

1. Able to construct logic for the algorithms designed using designing techniques.
2. Able to do posterior analysis of the algorithms.
3. Able to debug the algorithms.
4. Modify to improve performance of the algorithms.
5. Able to test and profile the algorithms.

Course Code: CA Lab-VI

**LAB on Web Designing
(HTML/XML/CSS/Navigation)**

Total Marks: 100

Course Objectives:

Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.

Become familiar with graphic design principles that relate to web design and learn how to implement these theories into practice.

Develop skills in analysing the usability of a web site.

Understand how to plan and conduct user research related to web usability.

Learn the language of the web: HTML, XML, and CSS.

Learn techniques of responsive web design, including media queries.

Learn CSS grid layout and flexbox.

Be able to embed social media content into web pages.

Demonstrate following Assignments with appropriate HTML tags and CSS :

1. Design a web page for clickable table of contents (TOC) of your first semester syllabus. Every click should display detailed content.
2. Design a web page for online books catalog.
3. Design a first page of online newspaper.
4. Design a web page for lecture schedule of current semester.
5. Demonstrate various CSS features
6. Design a web page that displays student feedback form for a scheduled course.
7. Design a web page that allows mp3 downloads.
8. Design a web page for nursery for plant selection using XML.
9. Design a web page for restaurant/ice cream parlor using XML.
10. Create a dictionary database using XML.
11. Design a narrative document using XML.
12. Display XML file using CSS
13. Display XML file of your choice using XSL
14. Display XML file using DSO data binding
15. Demonstrate Event Handling.
16. Draw Circle, Rectangle, Line, Ellipse and Polygon using SVG.
17. Create a Drawing Application with Canvas using HTML5.
18. Display Audio & Video playlist on browser using HTML5.
19. Display your location and location's coordinates of Longitude and latitude on Google Map.
20. Create a web page for shopping cart using Drag and Drop Events.
21. Design a web page which demonstrates Web Workers feature.

Course Outcome:

Select and apply markup languages for processing, identifying, and presenting of information in web pages.

Use scripting languages and web services to transfer data and add interactive components to web pages.

Create and manipulate web media objects using editing software.

Incorporate aesthetics and formal concepts of layout and organization to design websites that effectively communicate using visual elements.

Semester-IV

Course Code: CA-401

**High Performance Computing
Paradigms and Applications**

Clock Hours: 60

Total Marks: 100

Course Objectives:

- 1) The objective of this course is to provide students with the 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.
- 3) To provide knowledge of cloud concepts capabilities across the various cloud service models which includes IaaS, PaaS and SaaS.
- 4) To learn how to use Cloud Services, implementation of Virtualization, Task Scheduling algorithms.
- 5) Broadly educate students to know the impact of engineering on legal and social issues involved.

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.

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:

- 1) Analyze the Cloud computing setup with its vulnerabilities and applications using different architectures.
- 2) Apply and design suitable Virtualization concept, Cloud Resource Management.
- 3) Assess cloud Storage systems and Cloud security, the risks involved, its impact and develop cloud application.
- 4) Broadly educate to know the impact of engineering on legal and societal issues involved in addressing the security issues of cloud computing.
- 5) Assessment of own organizations needs for building capacity and training in cloud computing related IT areas

Course Code: CA-402

Web Application Technology

Clock Hours: 60

Total Marks: 100

Course Objectives:

Students will learn:

- 1) .Net Framework
- 2) Creating ASP.Net web applications using standard .net controls.
- 3) Develop database applications using ADO.Net
- 4) Use Web Services and develop simple and complex applications using .Net framework
- 5) Develop a data driven web application.
- 6) Connecting to data sources and managing them.
- 7) Maintain session and controls related information for user used in multi-user web applications
- 8) Understand the fundamentals of developing modular application by using object oriented methodologies

Unit-I

[10] Max Marks:15

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,.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-II

[15] Max Marks:25

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-III

[20] Max Marks:25

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(), 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-IV

[15] 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:

d. Website URLs

<https://www.asp.net/>

<http://asp.net-tutorials.com/>

e. 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 the course, student will be able to:

- 1) design web applications 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-403

Machine Learning

Clock Hours: 60

Total Marks: 100

Course Objectives:

- i. The course introduces next paradigm of computing i.e. beyond procedural and object-oriented instructional model to the students.
- ii. The course gives understanding of fundamentals of Machine Learning such as its types, applications and other preliminaries.

- iii. Course also introduces performance evaluation measures and metrics.
- iv. Course gives fair idea about all important techniques of Machine Learning such as Classification, Regression and Clustering.
- v. Course aims to discuss various classifiers and prevailing algorithms for clustering and regression technique to greater depth.
- vi. 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
Linear regression, 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:

- i. The course will empower students to work on the next generation computing platform.
- ii. The course will provide in-depth knowledge of various facets of Machine Learning methods/techniques and algorithms.
- iii. Students can envisage practical application of Machine Learning to Business and Research Computational problems.
- iv. Knowledge of Machine Learning will make student job-ready for Machine Learning based product development firms

Course Code: CA-404 Computer Graphics and Digital Image Processing

**Clock Hours: 60
Total Marks: 100**

Course Objectives:

1. To equip students with the fundamental knowledge and basic technical competence in the field of Computer Graphics and Digital Image Processing.
2. To provide an understanding of how to scan convert the basic geometrical primitives, basic principles of 2 and 3- dimensional computer graphics.
3. To be able to discuss the application of computer graphics concepts in the development of information visualization, and business applications.
4. Give an in-depth knowledge about the basic theory and algorithms related to Digital Image Processing.
5. Provide awareness about the current technologies and issues specific to Digital Image Enhancement, Restoration, Segmentation, Color Image Processing, and Morphological Image Processing.

Unit-I

[08] Max Marks:12

Introduction to Computer Graphics and Output Primitives: Overview of Computer Graphics, Its Applications and Software, Representation in Graphics, Vector Graphic Display, Raster Graphics Display, I/O Devices, Representing Image, Straight Line, Line Drawing Algorithm, DDA, Bresenham's Line Algorithm, Circle-generating algorithm, Ellipse-generating Algorithm, Polygon Filling Algorithm.

Unit-II

[06] Max Marks:12

Two-Dimensional Transformation: Matrix and transformation, 2D Basic transformation, Homogeneous coordinates, Translation, Scaling and Rotation of straight line or polygon, Combined Transformation, Rotation about an arbitrary point/line, Reflection and Shearing Transformation, Viewing Transformation, Clipping, Cohen-Sutherland line clipping.

Unit-III

[06] Max Marks:12

Three-Dimensional Transformation :Introduction, Matrix representation of 3D Transformation, 3D Translation, Scaling, Rotation, Composition of 3D Transformation, Projection, Orthographic, Isometric, Oblique Projection, Perspective Projection, One-Two-Three point perspective Projection.

Unit-IV

[08] Max Marks:12

Introduction to Digital Image Processing & Applications: 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-V

[10] Max Marks:15

Image Enhancement: 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-VI

[10] Max Marks:15

Image Restoration and Color Image Processing: A Model of the Image Degradation/Restoration Process, Noise Models, Restoration in the Presence of Noise Only-Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear, Position-Invariant Degradations, Estimating the Degradation Function, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Color Fundamentals, Color Models, Pseudocolor Image Processing, Basics of Full-Color Image Processing, Color Transformations, Smoothing and Sharpening.

Unit VII. Morphological Image Processing & Segmentation

[06] Max Marks:12

Detection of Discontinuities, Edge linking & Boundary Detection, Thresholding, Region based segmentation Laplacian of Gaussian, Derivative of Gaussian, Canny Edge Detection, Morphological operation: Dilation erosion, Opening & Closing, Basic Morphological Algorithm, Image representation schemes.

Unit VIII. MATLAB Image processing toolbox

[06] Max Marks:10

Introduction to MATLAB, Matrix Operations, Introduction to Image Processing Tool Box, Image Read & Write, Filters (spatial and frequency domain), Image Restoration and Reconstruction, Morphological Operations, Edge Detection and linking, Segmentation

References:

1. Amarendra N Sinha, Arun D. Udai, (2008). Computer Graphics, TMH publication ISBN-13 : 978-0-07-063437-4.
2. D. Hearn and M. Pauline Baker, Computer Graphics (C Version), Pearson Education, 2nd Edition ISBN-13: 978-0135309247
3. D. F. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, 2nd Edition, McGraw-Hill International Edition, ISBN-13:978-0-07-0486775
4. R.C.Gonzalez & R.E.Woods, Digital Image Processing, Pearson Education, 3rd edition, ISBN. 13:978-0131687288
5. S. Jayaraman Digital Image Processing TMH (McGraw Hill) publication, ISBN-13:978-0-07-0144798

6. Gonzalez, Woods & Steven, Digital Image Processing using MATLAB, Pearson Education, ISBN-13:978-0130085191

Course Outcome:

1. Developed scientific and strategic approach to solve complex problems Computer in the domain of Computer Graphics and Digital Image Processing.
2. Demonstrated 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. Built the mathematical foundations for digital image representation, image acquisition, image transformation, image enhancement and restoration.
4. Developed a theoretical foundation of fundamental concepts of digital image processing.
5. Exposed students to MATLAB Image Processing Toolbox.

Course Code: CA-405(A)

**Accounting and Management
Control**

*ClockHours:60
Total Marks:100*

Course Objectives:

1. This course aims at introducing the basic accounting procedures and financial management processes.
2. To maintain full and systematic records of business transactions
3. To ascertain profit or loss of the business
4. To depict financial position of the business
5. To provide accounting information to the interested parties
6. Cost Accounting objectives are as follows: (i) Ascertainment of cost, (ii) Determination of selling price, (iii) Cost control and cost reduction, (iv) Ascertaining the profit of each activity, (v) Assisting management in decision-making.
7. A management Accountant assists management in performing various functions by way of collecting, analysing, interpreting and presenting all accounting information which is useful to the management.
8. It also focuses on the computerised implementation of the various accounting principles discussed at different levels

Unit-I

[05] Max Marks:10

Principles of Accounting - Concepts - Conventions – Principles, methods of accounting, accounting systems, types of accounts.

Unit-II

[15] Max Marks:20

Financial Accounting –books of accounts, Journal, ledger, trial balance, Financial Statements.

Unit-III [07] Max Marks:10
Ratio Analysis, meaning, importance, scope, limitations, types of variances, calculation of variances.

Unit-IV [06] Max Marks:10
Cost Accounting - Concept of Cost, Elements of Cost, cost classification, methods and techniques of costing.

Unit-V [06] Max Marks:10
Fund Flow Statement, Concept, importance, Simple Problems, Cash Flow Statement, Concept, Advantages, Simple Problems.

Unit-VI [06] Max Marks:10
Standard Costing: concept, types of variances, computation of material cost variances, Labor Cost variances, and overhead cost variances.

Unit-VII [05] Max Marks:10
Marginal Costing, importance, Marginal Costing Vs Absorption Costing, importance, Break-even Analysis, break even chart.

Unit-VIII [05] Max Marks:10
Budgeting and Budgetary Control, Benefits, Zero Base Budgeting, Traditional Budgeting Vs Zero Base Budgeting, types of budget - fixed, flexible, cash budget.

Unit IX: [05] Max Marks:10
Responsibility Accounting: Responsibility Centre's, Types of Responsibility Centre's(Cost, Revenue, Profit, Investment), Responsibility, Performance Reporting: Measuring Divisional Performance, Non – Financial Measures of Performance.

References:

1. Dr. Sakshi Vasudeva, (2010),” Accounting For Business Managers”, Himalaya Publishing House. ISBN-10: 8184885636
2. Dr. Jawahar Lal, (2017) “Accounting For Management”, Himalaya Publishing House, ISBN 81 78663856
3. Bhattacharya, S.K. and Dearden, John. (1997), “Accounting for Management : Text and Cases“,3rd edition, Vikas Publishing House ISBN 10- 8125905537
4. Chadiwick.(1996),” The Essence of Financial accounting”, Prentice Hall, ISBN-10: 0133565106

5. Horngren, Sundem and Selto (1993), "Introduction to management accounting", 9th edition, Prentice Hall, ISBN- 0134820924

Course Outcome:

1. The major thrust of MCA student is to develop applications.
2. The student gets the knowledge of accounting procedures and financial management processes.
3. The various accounting principles studied at different levels can be used for automation of business.
4. To create entrepreneur in the field of information technology.

Course Code: CA-405(B)

Data Analytics

Clock Hours: 60

Total Marks: 100

Course Objectives:

This course will cover fundamental algorithms and techniques used in Data Analytics. The statistical foundations will be covered first, followed by various machine learning and data mining algorithms. Technological aspects like data management (Hadoop), scalable computation (MapReduce) and visualization will also be covered. In summary, this course will provide exposure to theory as well as practical systems and software used in data analytics.

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 completing this course, you will learn how to:

- 1) Find a meaningful pattern in data
- 2) Graphically interpret data

- 3) Implement the analytic algorithms
- 4) Handle large scale analytics projects from various domains
- 5) Develop intelligent decision support systems

Course Code: CA LAB-VII

**LAB on Computer Graphics and
Digital Image Processing**

Total Marks: 100

Course Objectives:

1. The student will gain a deeper knowledge about a chosen field of computer graphics and image processing while working on one of the more complex projects solved in the Laboratory.
 2. To implement line, circle and ellipse drawing algorithms and 2 and 3-dimensional geometric transformations using C++.
 3. To be able to design and develop the programs for viewing and clipping on graphical objects.
 4. To introduce MATLAB to implement the complex algorithms of Digital Image Processing.
 5. Provide hands-on experience to process digital images and expose students to MATLAB Image Processing Toolbox for Digital Image Enhancement, Restoration, Segmentation, Color Image Processing, and Morphological Image Processing.
1. Line drawing algorithm (DDA and Bresenham's Line Algorithm)
 2. Circle drawing algorithm
 3. Ellipse drawing algorithm
 4. Polygon filling algorithm
 5. Windowing and clipping algorithm (Point, line and polygon clipping)
 6. Composite 2-D transformation, (rotation, scaling & reflection)
 7. 3-D geometric transformation (rotation, scaling & reflection)
 8. Introduction to Image Processing Toolbox
 9. 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.
 10. Implement different interpolation techniques using MATLAB/ SciLab
 11. Read an image, plot its histogram then do histogram equalization. Comment about the result.
 12.
 - 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.'
 13. Implement various Smoothing spatial filter.

14. Read an image and apply
 1. Gaussian 3x3 mask for blurring
 2. High pass filter mask with different masks
 3. Laplacian operator with center value positive and negative
 4. High boost filtering.
15. Write a program to implement various low pass filters and high pass filter in frequency domain.
16. Write a program for erosion and dilation, opening & closing using inbuilt and without inbuilt function.
17. Implement and study the effect of Different Mask (Sobel, Prewitt and Roberts)
18. Implement various noise models and their Histogram
19. Implement inverse filter and wiener filter over image and comment on them

Course Outcome:

1. Developed scientific and strategic approach to solve complex problems Computer in the domain of Computer Graphics and Digital Image Processing using C++ and MATLAB respectively.
2. Implemented 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. Exposed students to MATLAB and Image Processing Toolbox.
4. Used various tools in MATLAB to implemented image transformation, image enhancement in spatial and frequency domain.
5. Developed the programs on various digital image processing techniques.

Course Code: CA LAB-VIII

**LAB on High Performance
Computing (HPC) and Web
Application Technology**

Total Marks: 100

Course Objectives:

- To expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.
- To provide knowledge of cloud concepts capabilities across the various cloud service models which includes IaaS, PaaS and SaaS.
- Set up a programming environment for ASP.net programs.
- Configure an asp.net application.
- Creating ASP.Net applications using standard .net controls.
- Develop a data driven web application.
- Connecting to data sources and managing them.
- Maintain session and controls related information for user used in multi-user web applications

- Understand the fundamentals of developing modular application by using object oriented methodologies

High Performance Computing

Practical's on HPC

The syllabus of Practical on HPC will be prepared before the beginning of the Fourth Semester by the Instructors based on the content of CA-401 Theory course. This will cover demonstration of use of IaaS, PaaS and SaaS along with some case studies such as Amazon EC2, Amazon S3, Aneka, Google AppEngine.

Web Application Development Technology Practical Assignments

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

Course Outcome:

- Apply and design suitable Virtualization concept, Cloud Resource Management.

- Assess cloud Storage systems and Cloud security, the risks involved, its impact and develop cloud application
- Successful students will be able to design web applications using ASP.NET
- Successful students will be able to use ASP.NET controls in web applications.
- Successful students will be able to debug and deploy ASP.NET web applications
- Successful students will be able to create database driven ASP.NET web applications and web services.

Semester- V

Course Code: CA-501

Compiler Construction

Clock Hours: 60

Total Marks: 100

Course Objectives:

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. Dhamdhare 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:

Understanding of basic structure of compiler, concepts and terminology in programming languages, lexical analysis, finite state techniques, scanner generator, parsing, kinds of parsers, designing lexical analyzer, scanner and parsers, principal ideas with intermediate code generation, optimizations.

Understanding of all concepts essential to design compiler in general for programming languages.

Course Code: CA-502

Software Engineering

Clock Hours: 60

Total Marks: 100

Course Objectives:

- 1) Knowledge of basic Software Engineering methods, practices and their appropriate application.
- 2) Describe software engineering layered technology and Process frame work which includes general understanding of software process models.
- 3) Understanding of software requirements, SRS documents, role of project management which includes planning, scheduling, risk management and Quality of product.
- 4) Describe data models, functional models and behavioral models.
- 5) Understanding of implementation issues such as modularity, coding standards, verification and validation which includes software testing approaches.

Unit-I

[10] Max Marks:10

Introduction and Process Models: Nature of Software, Software Engineering the process, Software Myths. Process Models: Generic process model, Prescriptive process models, Specialized process models, Unified process, Personal and Team process model, Process Technology, Product and Process. Agility, cost of change, Agile process, Extreme Programming, Agile Process models: Adaptive Software development, Scrum, Dynamic system development model, Crystal, Feature Driven development, Lean Software development, Agile modelling, Agile Unified process. Tool set for Agile process

Unit-II [10] Max Marks:15

Requirement Analysis and Modelling: Requirement Engineering, Establishing Groundwork, Eliciting Requirements Developing Use cases, Building Requirement model, Negotiating and Validating requirements. Requirement analysis, Scenario based modelling, UML models that supplements use case, Data modelling concepts, class based modelling. Requirement Modelling strategy, Flow oriented modelling, Creating Behavior model, Pattern for Requirement modelling.

Unit-III [08] Max Marks:15

Quality Assurance and Change Management: Elements of SQA, SQA Tasks, Goal and Metrics, Formal approaches to SQA, Software Reliability, ISO 9000 Quality standards, SQA Plan. Software Configuration Management, SCM Repository, SCM process

Unit-IV [11] Max Marks:20

Design Concept: Design process, Design Concept: Abstraction, Architecture, Pattern, Separation of concept, Modularity, Information hiding, Functional independence, Refinement, Aspects, Refactoring. Design Model: Data design element, Architectural design element, Interface design element, Component level design element, Deployment level design element.

Unit-V [11] Max Marks:20

Architectural and Component Level Design: Software Architectures, Architectural Genres, Architectural styles, Architectural design, Accessing alternatives Architectural design, Architectural mapping using dataflow. Introduction to component, Designing class based component, Conducting component level design, Designing traditional component, component based development.

Unit-VI [10] Max Marks:20

Software Testing: Strategic approach to software testing, Test strategies for conventional software, Validation Testing, System testing, Software testing fundamentals, Internal and external view of testing, White box testing, Basic path testing, Control structure testing, Black box testing, model based testing, Testing for specialized Environment, Architectures and applications.

References:

1. R. S. Pressman(2010), “Software Engineering: A Practitioner’s Approach”, McGraw-Hill International Edition, Seventh Edition, 2010, ISBN:978-007-126782-3.
2. Pankaj Jalote(2010), “Software Engineering: A Precise Approach”, Wiley India Pvt. Limited, 2010 ISBN: 978-81-265-2311-5.
3. K. K. Aggarwal and Yogesh Singh(2008), “Software Engineering”, Third Edition, New Age International Publishers, 2008, ISBN:978-81-224-2360-0.

Course Outcome:

After completing the course the students will be able to:

1. Basic knowledge and understanding of the analysis and design of complex systems.
2. Apply software engineering principles and techniques, to develop, maintain and evaluate large-scale software systems.
3. To produce efficient, reliable, robust and cost-effective software solutions.
4. To communicate and coordinate competently by listening, speaking, reading and writing English for technical and general purposes.
5. Ability to work as an effective member or leader of software engineering teams and manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals.

Course Code: CA-503

Natural Language Processing

Clock Hours: 60

Total Marks: 100

Course Objectives:

- i. 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.
- ii. 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.
- iii. Course provide knowledge of different approaches/algorithms for carrying out NLP tasks.
- iv. Course 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 and 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. Indurkha, 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:

- 1) Students will get idea about know-hows, issues and challenge in Natural Language Processing and NLP applications and their relevance in the classical and modern context.
- 2) Student will get understanding of Computational techniques and approaches for solving NLP problems and develop modules for NLP tasks and tools such as Morph Analyzer, POS tagger, Chunker, Parser, WSD tool etc.
- 3) Students will also be introduced to various grammar formalisms, which they can apply in different fields of study.
- 4) Students can take up project work or work in R&D firms working in NLP and its allied areas.

Course Code: CA-504

Mobile Application Development

Clock Hours: 60

Total Marks: 100

Course Objectives:

1. Understand basics of mobile application development.
2. To introduce Android platform and its architecture.
3. To learn activity creation and Android UI designing.
4. To be familiarized with Intent, Broadcast receivers and Internet services, SQLite Database and content providers.
5. To integrate multimedia, camera and Location based services in Android Application.

Unit- I

[10] Max Marks:10

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 [12] Max Marks:15

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:20

Creating Applications, activities and User Interfaces: What Makes an Android Application?, Introducing the Application Manifest. Using the Manifest Editor. The Android Application Life Cycle. Understanding Application Priority and Process States. Externalizing Resources. A Closer Look at Android Activities. Fundamental Android UI Design. Introducing Views. Introducing Layouts and fragments, Using Adapters, Creating New Views.

Unit-IV [12] Max Marks:20

Intents, Broadcast Receivers, and the Internet: Introducing Intents, Creating Intent Filters and Broadcast Receivers, Using Internet Resources. Introducing Dialogs and Action Bars, Creating and Using Menus.

Unit-V [06] Max Marks:15

Working in the Background: Introducing Services, Using Background Worker Threads, Using Toast, Introducing Notifications, Using Alarms.

Unit-VI [12] Max Marks:15

Data Storage, Retrieval, and Sharing: FILES, SAVING STATE, AND PREFERENCES: Saving Application Data, Creating and Saving Shared Preferences, Retrieving Shared Preferences, Introducing the Preference Framework and the Preference Activity, Persisting the Application Instance State, Including Static Files as Resources, Working with the File System. **DATABASES AND CONTENT PROVIDERS:** Introducing Android Databases, SQLite Databases, Content Providers, Introducing SQLite, Content Values and Cursors, Working with SQLite Databases, Creating Content Providers.

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:

1. Able to compare android with other smartphone OS and desktop OS.
2. Able to understand software stack of android OS.
3. Able to understand Activity lifecycle and management.
4. Activity creation and manipulation with effective UI design.
5. Understand use Intent, Broadcast receivers and Internet services.
6. Effectively use SQLite Database and content providers, multimedia, camera and Location based services in Android Application.
7. Demonstrate their ability to deploy software to mobile devices.

Course Code: CA-505 (A) CMS: Drupal Website Development Framework

**Clock Hours: 60
Total Marks: 100**

Course Objectives:

- 1) Distinguish between various content management systems capabilities
- 2) Create and customize contents for web pages
- 3) Create lists, views in drupal
- 4) Identify appropriate plugins, modules and themes for an application
- 5) Create websites using Wordpress and Drupal

Unit-I

[05] Max Marks:05

PHP Overview: PHP syntax highlights, PHP arrays, Superglobal Arrays, PHP and MySQL

Unit-II

[08] Max Marks:10

CMS: Definition, Advantages, Challenges. Survey of CMS systems and their features and comparison.

Drupal Introduction and Installation: History, Drupal community, Professional support , The Drupal association, Drupal Installation: single Drupal website, Installation Profiles,

Installing , Acquia's stack installer, Don't Hack Core, Core Files , Sites Folder, Inheritance and Overrides.

Unit-III [07] Max Marks:15

Drupal Website Development: Adding Content , Differences between Articles and Pages , Modifying the Defaults, Modifying your Menus, Adding Blocks to your website, Getting in contact, Adding the Contact Form to Your Main Menu, Summing Up the Contact Form , Exploring your site's permissions, Creating a members-only site, Exploring Roles, Wrapping up Users and Permissions.

Unit-IV [10] Max Marks:20

Administration-Configuration, Modules, and Reporting: Configuration, System — Site Information, Regional and Language — Settings, Media, Media — File System, Temporary Directory , PHP File Size Limitations , Media — Image Toolkit, Web Services, Web Services — RSS Publishing, Web Services — Feed Aggregator, Content Authoring — Text Formats, Explore the filtered html text format, Choosing Roles and the Importance of the Order.

Unit-V [10] Max Marks:20

Blocks, Menus, Themes and Content: Blocks, Block Visibility, Block Visibility Using PHP, Menus, url aliases aka custom paths, Themes, Theme Global Settings, Content, Nodes, Creating Custom Content types, Submission Form Settings, Publishing Options, Display Settings, Comment Settings, Menu Settings, Adding an Image Upload Field, Multiple Fields for Multiple Images? Permissions, Digging Deeper into Fields, Content Construction Kit (CCK), Views, Content Moderation, Comments, Permissions, WYSIWYG.

Unit-VI [10] Max Marks:15

User Management: Creating User Accounts, Creating Accounts, Canceling Accounts, Setting Up Account E-Mails, using OpenID, Creating Roles and Permissions, Creating Custom User Profiles, User Sessions.

Triggers, Actions, Workflow, and Rules: Triggers and Actions, Workflow, Rules, Features of the Rules Module, Conditions, Scheduler and Rule Sets, Input Evaluators.

Unit-VIII [10] Max Marks:15

Views: Overview of Views, The Frontpage View, Displays and Style Settings, Creating a Photo Gallery, Arguments, Exposed Filters, Relationships and Permissions, Caching, Advanced Views.

Theming: Installing Themes, Configuring Themes, Custom Theming, Theme Functions, .info file, Regions and Template Files (.TPL.PHP), Swapping Theme Functions And Template Files, Theme Registry, Theme Engines.

References:

1. Beginning Drupal, Jacob Redding ,Wiley Publishing, Inc., 2010,ISBN:978-0-470-43852-7
2. Beginning PHP 5.3, Matt Doyle , Wiley Publishing, Inc., 2010, ISBN: 978-0-470-41396-8
3. Learning PHP and MySQL , Michele E. Davis ,Jon A. Phillips , 2nd edition, Published by O'Reilly Media, Inc., August 2007, ISBN-10: 0-596-51401-8 | ISBN-13: 978-0-596-51401-3.

Auxiliary Resources:

- a) www.wiley.com , <https://www.tutorialspoint.com/drupal/> ,
<https://www.tutorialspoint.com/php/>
- b) https://www.youtube.com/watch?v=-DYSucV1_9w&list=PLtaXuX0nEZk9MKY_CIWcPkGtOEGyLTyCO ,
<http://getlevelten.com/tutorial/course/getting-started-drupal-7> ,
<https://www.tutorialspoint.com/listtutorial/PHP-Getting-Started-Video-Tutorial/5443> .

Course Outcome:

After completing this course the students will be able to

- 1) Compare and contrast between various content management systems
- 2) Enable and configure interactive capabilities
- 3) Demonstrate critical thinking skills to design and create different contents
- 4) Design and create content management system based websites

Course Code: CA-505(B)

Ruby on Rails

Clock Hours: 60

Total Marks: 100

Course Objectives:

Develop programming skills and logical ability with the use of Ruby Programming language platform.

Understanding web application development using Rails framework.

Unit-I

[08] Max Marks:12

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.

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:

Familiar with Ruby Programming language by understanding 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.

Familiar with web application development using Rails framework.

Course Code: CA LAB-IX

**LAB on Android
Programming**

Total Marks: 100

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.
- 3) To learn how to use intents to broadcast data within and between Applications.
- 4) To use Content providers and Handle Databases using SQLite.
- 5) To introduce Android APIs for Camera and Location Based Service.
- 6) To discuss various security issues with Android Platform.

1. Create “Hello World” application. That will display “Hello World” in the middle of the screen in the red color with white background with change in fonts & styles of text.
2. Create List with string taken from resource folder (res>>value folder). On changing list value change image.
3. Create android UI such that, one screen have radio button of the types of cars. On selecting any car name, next screen should show car details like: name, company name, and images if available, show different colors in which it is available.
4. Create android application that will display toast (Message) on android life cycle stages.
5. Create the application that will change color of screen based on selected option from the menu.
6. Create android application with login module. (Check username & password) on successful login, go to next screen. And on falling login, alert user using Toast.
7. Create android application that will display toast (Message) on specific interval time.
8. Create the android application that calls 3 native applications using 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. Create android application to take a picture using native application.
11. Create the android application that will send SMS using your android application.
12. Create android background application that will open activity on specific time.
13. Create the android application that will demonstrate shared preferences.
14. Create the android application that will call maps using android application.
15. Create android application to make Insert, Update, Delete and retrieve operation on employee database.

Course Outcome:

- 1) Design and Implement User Interfaces and Layouts of Android App.
- 2) Use Intents for activity and broadcasting data in Android App.
- 3) Design and Implement Database Application and Content Providers.
- 4) Experiment with Camera and Location Based service. Develop Android App with Security features.

Course Code: CA LAB-X

LAB on Drupal Framework/

Total Marks: 100

LAB on Ruby on Rails

LAB on Drupal Framework

Course Objectives:

Drupal is a free and open source Content Management System (CMS) that allows organizing, managing and publishing your content. This reliable and secure CMS is built on PHP based

environment and powers millions of applications and websites. This course will teach student the basics of Drupal using which they can create a blog or a website with ease.

1. Installation of Drupal.
2. Demonstrate menu Management in Drupal.
3. Write an application to create Blocks & Regions in Drupal.
4. Write an application to create Themes and Layouts in Drupal.
5. Demonstrate how to create Front Page & Static Pages in Drupal.
6. Demonstrate how to create Blogs & articles in Drupal.
7. Demonstrate how to create, modify, delete and publish content in Drupal.
8. Demonstrate User Management in Drupal.
9. Demonstrate URL Alias in Drupal.
10. Demonstrate Triggers and Actions in Drupal.
11. Demonstrate Views in Drupal.
12. Create a new custom theme with CSS in Drupal.

Course Outcome:

After completing this course, student will find themselves at a moderate level of expertise in developing websites using Drupal.

LAB on Ruby on Rails

Course Objectives:

- Develop and flourish programming and logical skill by practicing assignments.

Ruby and Rails 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
circle
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. Demonstrate Inheritance in Ruby by building a superclass called Bird from which our Duck, Goose, and Owl classes will derive their functionality. (http://www.gotealeaf.com/books/oo_ruby/read/inheritance)
5. Demonstrate a Ruby programs which uses loops like, each, times, do loop, etc. With having use of operators & exceptions which cause to break loop like divide by zero, etc.
6. Write a Ruby script which demonstrate use of blocks, lambda & proc.
7. Write a Ruby program which show duck typing, uses respond_to? Method.

8. Write a Ruby program which 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 various 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 nested 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 can'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:

The student will be able apply technical knowledge and perform specific technical skills, including:

- Ability to logically build program using syntactic structure and APIs of Ruby Programming Language.
- Ability to design simple web applications using Rails framework.

Semester- VI

Full Time Industrial Training

Course Code: CA-601

Full Time Industrial Training

Total Marks: 300

Course Objectives:

- To provide comprehensive learning platform to students where they can enhance their employ ability skills and become job ready along with real corporate exposure.
- To enhance students' knowledge in one particular technology.
- To Increase self-confidence of students and helps in finding their own proficiency.
- To cultivate student's leadership ability and responsibility to perform or execute the given task.
- To provide learners hands on practice within a real job situation.

Twelve credits shall be awarded to the Industrial Training/Project course, which will commence in the VIth semester and the final work and report will be completed at the end of VIth Semester of MCA course. 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 VIth semester of MCA course.

Course Outcomes:

- Capability to acquire and apply fundamental principles of Computers and information technology.
- Become master in one's specialized technology.
- Become updated with all the latest changes in technological world.
- Ability to communicate efficiently.
- Knack to be a multi-skilled IT professional with good technical knowledge, management, leadership and entrepreneurship skills.
- Ability to identify, formulate and model problems and find engineering solution based on a systems approach.
- Capability and enthusiasm for self-improvement through continuous professional development and life-long learning