GOVERNMENT ARTS COLLEGE (AUTONOMOUS)

KUMBAKONAM 612 002

Re - accredited With 'A' Grade by NAAC & Affiliated to Bharathidasan University

DEPARTMENT OF COMPUTER APPLICATIONS

(Effective for those admitted from 2020-2021 onwards)

SYLLABI

M.C.A.

GOVERNMENT ARTS COLLEGE (AUTONOMOUS), KUMBAKONAM

DEPARTMENT OF COMPUTER APPLICATION

Name of the Programme: **M.C.A – MASTER OF COMPUTER APPLICATIONS**

Academic Year of Revision: **2020-2021**

Programme Outcome (PO):

After the completion of this programme, the students will be able to

PO1: Apply knowledge from the course-discipline in all relevant fields of application due to employment

PO2: Suggest measures to the learner to augment their skills to solve problems of application areas

PO3: Become Computer scientists, entrepreneurs, Technical administrators, and also Technical faculty in institutional bases as teachers

PO4: Get employed in coveted positions of private and public sector firms and corporate establishments

PO5: Carry out research in Computer science and Applications arena

Programme Specific Outcome (PSO):

After the completion of this programme, the students will be able to

PSO1: Face the job market with high professional skills in area of application

PSO2: Suggest measures to solve problems related to application areas

PSO3: Acquire deep knowledge in computer science and computational skills so that they can face boldly to qualify in exams like NET/SET/GATE, UPSC and Banking type examinations

PSO4: Inculcate Computational Know-how in all aspects of approach and to tune skills thereof

PSO5: Pursue research in Computer Science and Computer application areas

GOVERNMENT ARTS COLLEGE (AUTONOMOUS), KUMBAKONAM - 612 001

Master of Computer Applications (M.C.A)

(Two Year course w.e.f 2020-2021)

Course Structure under CBCS

(Applicable to the candidates to be admitted during the academic year 2020-2021 and onwards)

Eligibility: A candidate who is a graduate in BCA (OR) graduate in any science / arts discipline / applied science with mathematics as a subject in graduate degree level (or) higher-secondary / plus two level.

Semester	Code	Course	Course Title	Ins. Hrs/ week	Credit	Exam Hrs	Marks		Total
							Int	Ext	
1		CC-I	Discrete Mathematics	4	3	3	25	75	100
		CC-II	Programming in C++	4	3	3	25	75	100
		CC-III	Data Structures and Algorithms	4	3	3	25	75	100
		CC-IV	Digital Electronics and Microprocessors	4	4	3	25	75	100
		CC-V	Operating System	4	3	3	25	75	100
		CP-I	Data Structures Lab using C++	5	3	3	40	60	100
		CP-II	Digital Electronics and Microprocessors Lab	5	3	3	40	60	100
			Total	**30**	**22**	-	-	-	**700**
2		CC-VI	Advanced Java Programming	4	3	3	25	75	100
		CC-VII	Database System Concepts	4	4	3	25	75	100
		CC-VIII	Graphics and Multimedia	4	3	3	25	75	100
		CC-IX	Data Communication And Networking	4	4	3	25	75	100
		CC-X	Management Information System	4	3	3	25	75	100
		CP-III	Advanced Java Programming Lab	5	3	3	40	60	100
		CP-IV	RDBMS (Oracle) Lab	5	3	3	40	60	100
			Total	**30**	**23**	-	-	-	**700**
3		CC-XI	Open Source Technologies (PHP and MySQL)	4	3	3	25	75	100
		CC-XII	Principles of Compiler Design	4	4	3	25	75	100
		CC-XIII	.NET Frame work	4	3	3	25	75	100
		EC-I	(A) Emerging Trends And Technology (B) Image Processing (C) Neural Network And Fuzzy Systems	4	3	3	25	75	100
		EC-II	(A) Probability and Statistics (B) Numerical Methods and Applications (C) Operations Research	4	3	3	25	75	100
		CP-V	Open Source Technologies Lab	5	3	3	40	60	100
		CP-VI	.NET Lab	5	3	3	40	60	100
			Total	**30**	**22**	-	-	-	**700**
		CC-XIV	Software Engineering	4	3	3	25	75	100
		CC-XV	Data Science	4	4	3	25	75	100
		CC-XVI	Python Programming	4	3	3	25	75	100
		EC-III	(A) Artificial Intelligence & Expert	4	3	3	25	75	100

4			System (B) E-Commerce (C) Human Computer Interaction						
		EC-IV	(A) Parallel Computing (B) Grid Computing (C) Cloud Computing	4	3	3	25	75	100
		CP-VII	Data Analytics using Python Lab	5	3	3	40	60	100
		PW	**Project Work**	5	4	3	40	60	100
			Total	**30**	**23**	-	-	-	**700**
	Grand Total			-	**90**	-	-	-	**2800**

Course Type	Course	Credits
Core (Theory)	16	53
Core(Practical)	07	21
Core(Project Work)	1	4
Elective	4	12
Total	**28**	**90**

DISCRETE MATHEMATICS

OBJECTIVE: To Obtain Knowledge from the Basic elements and Derive the Discrete Digital Process from this Subject.

UNIT I: Mathematical Logic: Propositions, Connectives, Conditional and Bi conditional Proposition, Equivalence of Proposition, Algebra of Proposition, Truth table construction, Predicate Calculus, Quantifiers. Set Theory: Basic Concept And Notation, Set Operation.

UNIT II: Relations: Types of Relations, Some operations on Relations, Properties of Relations, Matrix Representation of Relations. Lattices: Principles of Duality, Properties of Lattices, Lattices as Algebraic Systems, Boolean Algebras, Subalgegra, Boolean Expressions and Boolean functions, Logic Gates Karnaugh Map Method, Functions: Definition, Types of Functions, Composition of Functions, Inverse of a Function, Recursive Functions, Hashing Function.

UNIT III: Algebraic Structures: Semi groups and Monoids, Groups, subgroups and Homomorphism. Residue Arithmetic, Fermat's Theorem, Euler's Theorem, Group Codes: Communication model of Error Correction, Parity Checker, Error Recovery in Group Codes.

UNIT IV: Graphs Theory: Basic Definition, Degree of a Vertex, Edge, Simple Graph, Matrix Representation of Graphs. Incidence Matrix, Adjacency Matrix, Huffman Adjacency Matrix. Paths, Cycles and connectivity. Graph Traversals: DFS and BFS, and Shortest Path Algorithms, Warshall's Algorithm.

UNIT V: Trees : Trees, Spanning Tree, Minimum Spanning Tree, Rooted and Binary Trees, Binary Tree, Tree Traversal, Expression Tree, Kruskal's Algorithm and Prim's Algorithm.

OUT COMES:

After completing this subject the student will be able to

- Gain Knowledge in mathematical logic and algebraic structure. It plays more role in computer science field in the area of Artificial Intelligence.
- And also Gain knowledge in Graph theory and Trees. It is more useful to develop Graph Algorithm. These Algorithms are used to solve graph theoretical concepts to solve the corresponding computer science application.

TEXT BOOK:

"Discrete Mathematics with Graph Theory and combinatorics", T. Veerarajan, Pub: Tata McGraw Hill.

REFERENCE BOOK:

. "Elements of Discrete Mathematics", C.L. Liu and Mohapatra, Pub: Tata McGraw Hill.

"Discrete Mathematical Structures With Applications to Computer Science", J.P. Tremblay and R. Manohar. Pub: Tata McGraw Hill.

PROGRAMMING IN C++

OBJECTIVE: The objective of course is to develop programming skills of students, using object oriented programming concepts, learn the concept of class and object using C++ and develop classes for simple applications.

UNIT I: Basic Concepts of Object- Oriented Programming - Benefits of OOP – Object Oriented Languages - Applications of OOP – Structure of C++ Program –Tokens and Expressions: Keywords, Basic Data Types, User Defined Data Type, Derived Data Type, Declaration of Variables, Dynamic Initialization of Variables, Reference Variables, Operators in C++, Scope Resolution Operator, Member Dereferencing Operators, Memory Management Operators, Type Cast Operator, Expressions and Implicit Conversions, Operator Precedence

UNIT II: Control Structures: Simple If Statement, If..Else Statement, Switch Statement, Do-While Statement, While Statement, For Statement - Functions in C++: Main Function, Function Prototyping, Call By Reference, Return By Reference, Inline Functions, Default Arguments, Const Arguments, Function Overloading

UNIT III: Classes and Objects: Specifying a Class, Defining Member Functions, Making an Outside Function Inline, Nesting of Member Functions, Private Member Functions, Arrays within a Class, Arrays of Objects, Objects as Function Arguments, Friendly Functions, Returning Objects, Const Member Functions – Constructors and Destructors: Constructors, Parameterized Constructors, Multiple Constructors in a Class, Constructors with Default Arguments, Copy Constructor, Dynamic Constructor, Destructors

UNIT IV: Operator Overloading and Type Conversions: Defining Operator Overloading, Overloading Unary Operators, Overloading Binary Operators, Manipulation of Strings Using Operators, Rules for Overloading Operators, Type ConversionsInheritance : Introduction, Defining Derived Classes, Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance

UNIT V: Pointers ,Virtual Functions and Polymorphism: Introduction, Pointers to Objects, This Pointer, Pointers to Derived Classes, Virtual Functions - Managing Console I/O Operations: C++ Streams and Classes, Unformatted I/O Operations, Formatted Console I/O Operations – Working with Files: Introduction, Classes for File Stream Operations, Opening and Closing a File, Detecting End-of-file, Sequential Input and Output Operations, Updating a File Random Access, Error Handling during File Operations

COURSE OUTCOMES:

- Identify importance of object oriented programming and difference between structured oriented and object oriented programming features.
- Able to make use of objects and classes for developing programs.
- Able to use various object oriented concepts to solve different problems.

TEXT BOOK: Balagursamy E, Object Oriented Programming with C++, Tata McGraw Hill Publications, Sixth Edition, 2013

REFERENCE BOOKS: Ashok Kamthane, Programming in C++, Pearson Education,2013.

*****

MCA - DATA STRUCTURES AND ALGORITHMS

OBJECTIVES: To introduce the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms.

UNIT – I: Introduction: History of algorithms – Definition, Structure and Properties of Algorithms – Development of an Algorithm – Data Structures and Algorithms – Data Structure-Definition and Classification. **Analysis of Algorithms:** Efficiency of Algorithms – Apriori Analysis – Asymptotic Notations – Time Complexity of an Algorithm Using O Notation – Polynomial Vs Exponential Algorithms – Average, Best and Worst Case Complexities – Analyzing Recursive Programs.

UNIT – II: Arrays: Introduction – Array Operations – Number of Elements in an Array – Representation of Arrays in Memory – Applications. **Stacks:** Introduction – Stack Operations – Applications. **Queues:** Introduction – Operations on Queues – Circular Queues. **Linked Lists:** Introduction – Singly Linked List – Circularly Linked List – Doubly Linked List – Multiply Linked List.

UNIT – III: Trees and Binary Trees: Introduction – **Trees:** Definition and Basic Terminologies – Representation of Trees – **Binary Trees:** Basic Terminologies and Types – Representation of Binary Trees – Binary Tree Traversals – Threaded Binary Trees. **Graphs:** Introduction – Definition of Basic Terminologies – Representations of Graphs – Graph Traversals.

UNIT – IV: Binary Search Trees and AVL Trees: Introduction – **Binary Search Trees:** Definition and Operations. **AVL Trees:** Definition and Operations. **File Organization:** Introduction – Files – Keys – Basic File Operations – Heap or Pile Organization – Sequential File Organization – Indexed Sequential File Organization – Direct File Organization.

UNIT – V:
Searching: Introduction – Linear Search – Transpose Sequential Search – Interpolation Search – Binary Search – Fibonacci Search – Other Search Techniques. **Internal Sorting:** Introduction – Bubble Sort – Insertion Sort – Selection Sort – Merge Sort – Shell Sort – Quick Sort – Heap Sort – Radix Sort.

OUT COMES: After completing this subject the student will be able to

- To describe the usage of various data structures
- To write programs using linked structures such as List, trees, and graphs
- To understand basic algorithms such as Binary Search, Fibonacci Search, heap sort, graph traversal-based, radix-based sorting, AVL trees.
- To design and apply appropriate data structures for solving computing problems
- To develop computer programs to implement different data structures and related algorithms
- To analyze algorithms and to determine algorithm correctness and time efficiency class
- To demonstrate various methods of organizing large amounts of data.
- To apply and implement learned algorithm design techniques and data structures to solve problems.
- To implement and use advanced data structures (dynamic hash structures, heaps, AVL and multiway search trees, radix-based search trees)

Text Book: "Data Structures and Algorithms – Concepts, Techniques and Applications" – **G . A . Vijayalakshmi Pai** (G A V PA I) – TATA McGraw – Hill, 2008.

Reference: "Data Structures and Algorithms" – Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman – Pearson Education, 2001.

DIGITAL ELECTRONICS AND MICROPROCESSORS

OBJECTIVE: To disseminate knowledge of digital principles, combinational and sequential logic fundamentals, to introduce microprocessor, its principles and its applications

UNIT I: Number Systems and Codes: Binary, Octal and Hexadecimal Number Systems- Conversion between number systems- Complements - Binary Arithmetic- Binary CodesBoolean Algebra and Logic Gates: AND, OR, NOT, NAND, NOR, XOR, XNOR gates –Truth tables- Applications of XOR gates- Fundamentals of Boolean Algebra- Booleanfunctions- Minterms and Maxterms.

UNIT II: Boolean Laws and Expressions: Laws and theorems of Boolean algebra- Demorgan'stheorems- the Universal building blocks – NAND and NOR gates - Canonical SOP and POS forms- Algebraicsimplification-Karnaugh Maps- SOP and POS Simplification- NAND / NORimplementation of Boolean expressions- Don't care conditions- Overlapping, Rollinggroups, eliminating redundant groups.

UNIT III: Combinational Logic Circuits: Half and FullAdders- Half and Full Subtractors- Parallel binary adder- Multiplexer & De-Multiplexer-Encoder & Decoder.Sequential Logic Circuits: NAND, NOR latches- SR Flipflop- JK Flipflop – Edgetriggering- PRESET and CLEAR inputs- Shift Register, - Intermediary to Binary Counters -Asynchronous Forms- BCD counter.

UNIT IV: Microprocessor architecture: Introduction- Intel 8085: ALU-Timing and Control unit – Registers-Data and Address Bus-Pin configuration-Intel 8085 instructions—Instruction cycle- Timing diagram- RISC and CISC processors.

UNIT V: Instruction Set for Intel 8085: Instruction and Data formats- Addressing modes- Status Flags- Intel 8085instruction groups.Assembly Language Programming: Addition- Subtraction- Decimal addition /subtraction- Complement Arithmetic- Shifting – Masking- concept of Arrays and

operation on array values- Sum of Series – Multiplication – Division- Multibyte addition / subtraction.

COURSE OUTCOME:

- Describe the various structures of various number systems and its application in digital design.
- Develop the appropriate truth table from a description of a combinational logic function.
- Implement combinational logic function described by a truth table using and/or/inv gates, muxes or ROMs, and analyze its timing behavior.
- Describe the operation and timing constraints for latches and registers.
- Design memory organization that uses banks for different word size operations.

TEXT BOOKS:

1. Thomas Bartee C, Digital Computer Fundamentals. TMH, 3rd Edition
2. Malvino and Leech, Digital principles and Applications, TMH, 2nd Edition
3. Badri Ram, Fundamentals of Microprocessors and Microcomputers, DhanpatRai Publications (P) Ltd, 4th Edition

OPERATING SYSTEMS

OBJECTIVES:

The student should be made to:

- Study the basic concepts and functions of operating systems.
- Understand the structure and functions of OS.
- Learn about Processes, Threads and Scheduling algorithms.
- Understand the principles of concurrency and Deadlocks.
- Learn various memory management schemes.
- Study I/O management and File systems.
- Learn the basics of Linux system and perform administrative tasks on Linux Servers.

Unit 1: OVERVIEW: Introduction: What is Operating Systems - Computer System Organization – Computer System Architecture - Operating System Structure - Operating System Operations . Operating System Structures: Operating System Services - User and Operating - System Interface - System Calls - Types of System Calls.

Unit 2: PROCESS MANAGEMENT: Processes: Process Concept - Process Scheduling. CPU Scheduling: Basic Concepts - Scheduling Criteria. Deadlocks: System Model - Deadlock Characterization - Methods for Handling Deadlocks - Deadlock Prevention - Deadlock Avoidance - Deadlock Detection - Recovery from Deadlock

.

Unit 3: MEMORY MANAGEMENT: Main Memory: Background - Swapping - Contiguous Memory Allocation – Segmentation. Virtual Memory: Background - Demand Paging.

Unit 4: STORAGE MANAGEMENT: Mass-Storage Structure: Overview of Mass-Storage Structure - Disk Structure - Disk Attachment - Disk Scheduling - Disk Management. File-System Interface: File Concept - Access Methods. File-System Implementation: File System Structure - File System Implementation.

Unit 5: PROTECTION AND SECURITY : Protection : Goals of Protection - Principles of Protection - Domain of Protection. Security : The Security Problem - Program Threats - Firewalling to Protect Systems and Networks – Computer Security Classifications. Case studies: The Linux System - Linux History - Design Principles.

OUTCOMES:

At the end of the course, the student should be able to:

- Design various Scheduling algorithms.
- Apply the principles of concurrency.
- Design deadlock, prevention and avoidance algorithms.

- Compare and contrast various memory management schemes.
- Design and Implement a prototype file systems.
- Perform administrative tasks on Linux Servers.

TEXT BOOK:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", 9th Edition, John Wiley and Sons Inc., 2012.

REFERENCES:

1. William Stallings, "Operating Systems – Internals and Design Principles", 7th Edition, Prentice Hall, 2011.
2. Andrew S. Tanenbaum, "Modern Operating Systems", Second Edition, Addison Wesley, 2001.
3. Charles Crowley, "Operating Systems: A Design-Oriented Approach", Tata McGraw Hill Education", 1996.
4. D M Dhamdhere, "Operating Systems: A Concept-Based Approach", Second Edition, Tata McGraw-Hill Education, 2007.

DATA STRUCTURES LAB USING C++

Objectives: Have a good overall understanding of basic data structures, their design and analysis. Know how to implement many of them in C++, and use them in other applications.

Write C++ programmes to implement the following:

1. SORTING:

    a. Bubble Sort
    b. Insertion Sort
    c. Selection Sort
    d. Heap Sort
    e. Quick Sort

2. SEARCHING:

    a. Linear Search
    b. Binary Search

3. Matrix Manipulations

4. Polynomial Addition & Multiplication

5. Operations on Stack and Conversion of expressions

6. Operations on Queue

7. Operations on Linked List

8. Operations on Doubly Linked List

9. Operations on Binary tree and Traversals

10. File Processing

Course Outcomes:

- Understand basic data structures such as arrays, linked lists, stacks and queues.
- Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data.
- Understand the use of various OOPs concepts with the help of programs.

DIGITAL ELECTRONICS AND MICROPROCESSORS LAB

OBJECTIVE: To make students know the fundamental principles behind digital electronicsincluding basic gates, combinational and sequential logic and to introduce microprocessor,its principles and its applications on practical scope.

EXPERIMENTS:

DIGITAL ELECTRONICS LAB:

1. Basic gates using IC- AND, OR, NOT, NAND, NOR, XOR, XNOR gates.
2. Universal building blocks – NAND Logic.
3. Combinational Logic Circuits: Half and Full Adders.
4. Combinational Logic Circuits:Half and Full Subtractors
5. Sequential Logic Circuits: NAND, NOR latches
6. JK Flip flop
7. Asynchronous counter

MICROPROCESSORS LAB-Assembly language programming:

8. Simple Addition / Subtraction
9. Decimal addition / subtraction
10. Complement Arithmetic
11. Shifting values / Masking values
12. Arrays operation – Addition
13. Sum of Series
14. Double Precision Arithmetic

COURSE OUTCOME:

- Construct basic combinational circuits and verify their functionalities.
- Apply the design procedures to design basic sequential circuits.
- Learn about counters and Shift registers.

TEXT BOOKS:

1. Thomas Bartee C, Digital Computer Fundamentals. TMH, 3rd Edition
2. Malvino and Leech, Digital principles and Applications, TMH, 2nd Edition
3. Badri Ram, Fundamentals of Microprocessors and Microcomputers, DhanpatRai Publications (P) Ltd, 4th Edition

ADVANCED JAVA PROGRAMMING

Objective:

To understand the development of enterprise applications using Java 2 Platform, enterprise Edition (J2EE), EJB technology enables rapid and simplified development of distributed, transactional, secure and portable applications based on Java technology.

Unit I:
JDBC Overview - Connection Class - MetaData Function - SQLException - SQL warning - Statement - ResultSet - Other JDBC Classes.

Unit II:
InetAddress - TCP/ IP client sockets - TCP/ IP server sockets - URL - URL Connection - Datagrams - Client/ Server application using RMI.

Unit III:
Bean Development Kit - Jar Files - Introspection - Design Pattern for properties, events and methods - Constrained Properties - Persistence – Customizers

Unit IV:
Life Cycle of Servlet - Generic Servlet - HTTP Servlet - Reading Initialization Parameters - Reading Servlet Parameters - Cookies - Session Tracking

Unit V:
JApplet - Button - Combo - Trees - Tables - Panes - AWT Classes - working with Graphics, Color and Font

Outcomes:
At the end of the course the student will

- Develop Swing-based GUI
- Develop client/server applications and TCP/IP socket programming
- Update and retrieve the data from the databases using SQL
- Develop distributed applications using RMI
- Develop component-based Java software using JavaBeans
- Develop server side programs in the form of servlets

TEXT BOOK(S):

1. Patrick Naughton & Herbert Schildt, "The Complete Reference: Java 2", Tata McGraw Hill, 1999. (Chapter - 18, 21, 24, 25, 26, 27)
2. Joseph Weber, "Using Java 2 Platform", Prentice Hall of India, 2000. (Chapter - 39, 40)

REFERENCE BOOK(S):

1. Deitel & Deitel, "Java How to Program", Prentice Hall, 5th Edition ,2002
2. Peter Haggar, "Practical Java: Programming Language Guide", Addison-Wesley Pub Co, 1st Edition, 2000
3. Bruce Eckel, "Thinking in Java", Pearson Eduction Asia, 2nd Edition, 2000

GOVERNMENT ARTS COLLEGE (AUTONOMOUS), KUMBAKONAM.

M.C.A

DATABASE SYSTEM CONCEPTS

OBJECTIVE: To provide the basic concepts of the Database systems including data models, Storage Structure, Normalization and SQL.

UNIT – I:

Introduction: Database-System Applications –Purpose of Database Systems – View of Data – Database Languages – Relational Databases – Database Design – Data Storage and Querying – Data Mining and Analysis – Database Architecture – Database Users and Administrators – History of Database Systems.

UNIT – II:

Relational Model: Structure of Relational Databases – Fundamental Relational Algebra Operations – Additional Relational Algebra Operations – Extended Relational Algebra Operations – Null Values – Modification of the Database. **Storage and File Structure:** File Organization – Organization of Records in Files – Data-Dictionary Storage.

UNIT – III:

SQL: Overview of the SQL Query Language – Data Definition – Basic Structure of SQL Queries – Set Operations – Aggregate Functions – Null Values – Nested Subqueries – Modification of the Database. **Advanced SQL:** SQL Data Types and Schemas – Integrity Constraints – Authorization. **Other Relational Languages:** The Tuple Relational Calculus – The Domain Relational Calculus.

UNIT – IV:

Database Design and the E-R Model: Overview of the Design Process – The Entity Relationship Model – Constraints - Entity-Relationship Diagrams – Weak Entity Sets. **Relational Database Design:** Features of Good Relational Designs – Atomic Domains and First Normal Form – Decomposition Using Functional Dependencies – Functional Dependency Theory – Decomposition Using Functional Dependencies – Decomposition Using Multivalued Dependencies.

UNIT – V:

Indexing and Hashing: Basic Concepts – B+-Tree Index Files – B-Tree Index Files – Static Hashing – Dynamic Hashing. **Transactions:** Transaction Concept – Transaction State – Implementation of Atomicity and Durability. **Concurrency Control:** Lock-Based Protocols – Timestamp-Based Protocols – Validation-Based Protocols. **Distributed Databases:** Homogeneous and Heterogeneous Databases – Distributed Data Storage – Distributed Transactions – Commit Protocols.

COURSE OUTCOMES:

- Describe the fundamental elements of relational database management systems

- Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.
- Design ER-models to represent simple database application scenarios
- Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.
- Improve the database design by normalization.

Text Book :

Database System Concepts, Fifth Edition, Abraham Silberschatz, Henry F. Korth and S. Sudarshan, McGraw-Hill – 2006.

References:

1. Database Systems: Models, Languages, Design and Application, Ramez Elmasri, Pearson Education 2014.
2. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", McGraw Hill, Third Edition 2004.

GRAPHICS AND MULTIMEDIA

Objective: To import knowledge on the graphics primitives, advanced concepts of computer graphics and to disseminate ideas of multimedia applications to post graduate students.

Unit I:
Overview of graphics systems: Video display devices – Raster scan systems – input devices – hard cope devices. Output primitives: Points and lines – line drawing algorithms – DDA algorithm – Bresenham's line algorithm – circle generation algorithm – Filled – boundary fill algorithm – flood fill algorithm.

Unit-II:
Attributes of output primitives: Line attributes – curve attributes – color and grayscale levels – area fill attributes – character attributes – bundled attributes – inquiry functions. Two dimensional geometric transformations: Basic transformation – matrix representation – composite transformation – other transformation.

Unit-III:
Two dimensional viewing: Window to viewport coordinate transformation – two dimensional viewing functions – clipping operations – point clipping – line clipping – text clipping. Graphical user interfaces and interactive input methods: Input of graphical data – input functions – interactive picture construction techniques.

Unit-IV:
Three dimensional concepts: Three dimensional display methods: Parallel projection – perspective projection. Three dimensional object representations: Polygon surface – curved lines and surfaces. Three dimensional geometric and modeling transformations: Translation – rotation – scaling – other transformation. Three diemensional viewing: Projections.

Unit-V: Computer Animation: Design of animation sequence – general computer animation functions – raster animation – computer animation languages – key frame systems – morphine – simulating accelerations – motion specification – direct motion specification – goal directed systems – kinematics and dynamics.

OUTCOMES:

Students should be able to: explain, discuss and solve simple problems in the basic representation and handling of multimedia data (images, audio and animation), and the basic components of a 3D-environments.

TEXT BOOK(S):

Computer graphics – Donald Hearn & M. Pauline Baker – Prentice – Hall of india private limited.

REFERENCE BOOK(S):

Newman William M, & Sproull Robert F, Principles of interactive computer graphics, Second edition, Tata –McGraw Hill, 1 (ISBN 0-07-463293-0)

DATA COMMUNICATION AND NETWORKING

Objective: This course aims to provide all the program, date and hardware is available to thestudents on the network without regard to the physical location of the resource and the users.

Unit I:
Introduction – Data Communications – Networks – Protocols and Standards. Network models–Layered Tasks – The OSI model – Layers in the OSI model – TCP/IP Protocol suite – Addressing.

Unit II:
The physical layer and media – Data and Signals: Analog and Digital – Periodic Analog signalsDigital signals – Transmission Impairment – Data rate limits. Transmission media: Guidedmedia – Unguided media. Switching: Circuit switched Networks – Datagram Networks – VirtualCircuit Networks – Structure of a switch.

Unit III:
Data link layer – Error detection and Correction: - Introduction – Block coding – Linear blockcodes – Cyclic codes – Checksum. Data Link Control: - Framing – Flow and Error control –HDLC – Point-to-Point protocols.

Unit IV:
Network layer IPV4 Addresses – IPV6 Addresses –Network layer Internet Protocol: Internetworking - IPv4-IPv6. Network Layer: Delivery, Forwarding, and Routing: Delivery, Forwarding.

Unit V:
Transport layer and Application layer Process-to-process Delivery – User Datagram Protocol (UDP) – TCP – SCTP Congestion control and Quality of Service:- Data Traffic – Congestion – Congestion Control – Examples – Quality of Service. Application Layer: Domain Name System(DNS) –Name Space – Domain Name System - DISTRIBUTION OF NAME SPACE.

Course Outcome:
CO1 Understand and explain the concept of Data Communication and networks, layered architectureand theirapplications.
CO2 Analyse and Set up protocol designing issues for Communication networks.
CO3 Evaluate data communication link considering elementary concepts of data link layer protocols for error detection and correction.
CO4 Apply various network layer techniques for designing subnets and supernets and analyse packet flow on basis of routing protocols.
CO5 Estimate the congestion control mechanism to improve quality of service of networking application
CO6 Understand and design application layer protocols and internet applications such as network security, Email and DNS,

TEXT BOOK(S):
Behrouz A.Forouzan, 2008, Data Communication and Networking, Fourth Edition, TataMcGraw Hill, New Delhi.

REFERENCE BOOK(S):
Tanenbaum, A.S., 2001, Computer Networks, PHI, New Delhi, 3rd Edition.
Halsall, Fred, 2001, Data communication Computer Network and Open System, 4thEdition, Pearson education.
Black, Uyless.D, 1997, Computer Networks: Practical Standards and Interfacing, 2ndEdition,PHI, New Delhi.
Comer Douglas. E, 1999, Computer Networking and Internets, 2ndEdition,PHI, New Delhi.

****

MANAGEMENT INFORMATION SYSTEM

Objective:

- Demonstrate basic computer literacy by recalling terminology and concepts related to hardware, software, and networks
- Communicate effectively with technical and non-technical colleagues
- Use MIS to enhance decision making
- Generate informational reports
- Build and/or interpret models for planning
- Organize, summarize, and analyze data

Unit I

IT Foundation Concepts: Computer Hardware – Software – Data Resource Management – Telecommunications and Networks

Unit II

IS in Business Applications: Information Systems and Technologies – Business applications, development and Management

Unit III

E-Business and E-Commerce: The e-Business enterprise – e-Commerce – e-Business systems – e-Business Decision Support

Unit IV

Development Process: Developing e-Business Strategies and Solutions

Unit V

Security and Ethics in E-Business: Security, Social and Ethical Challenges in e-Business Enterprise and Global Management of e-Business Real World Cases Icongo, yahoo, IBM, SAP, Oracle, Siebel, Wall Mart

Outcomes

- Describe managing the digital firm
- Evaluate the role of information systems in today's competitive business environment.
- Define an information system from both a technical and business perspective and distinguish between computer literacy and information systems literacy.
- Assess the relationship between the digital firm, electronic commerce, electronic business and internet technology.
- Identify the major management challenges to building and using information systems in organizations.

Text Book(s)

1. Management Information Systems, Sixth Edition, James A.O' Brien, Tata-McGraw Hill Edition.

References

1. Management Information Systems, Third Edition, Post and Anderson, Tata-McGraw Hill Edition.

2. Frontiers of Electronic Commerce, Kalakota and Winston, Pearson Education.

3. Information Technology for Management, Seventh Edition, Henry C.Lucas Jr., TMG Edition .

ADVANCED JAVA PROGRAMMING LAB

Objectives:

- Programming in the advanced Java programming language.
- Knowledge of object-oriented paradigm in the Java programming language.
- The use of Java in a variety of technologies and on different platforms.

1. Write an Applet which will play two sound notes in a sequence continuously use the play() methods available in the applet class and the methods in the Audio clip interface.
2. Create a Japplet using swing control, which will create the layout shown below and handle necessary events.

Format Enter your Name:

Enter your Age:

Select your s/w: * Oracle *Visual Basic *Java

Select your city : *Delhi *Mumbai *Chennai

OK Cancel

3. Use JDBC connectivity and create Table, insert and update data.
4. Write a program in Java to implement a Client/Server application using RMI.
5. Write a program in Java to create a Cookie and set the expiry time of the same.
6. Write a program in Java to create Servlet to count the number of visitors to a web page.
7. Write a program in Java to create a form and validate a password using Servlet.
8. Develop a Java Bean to demonstrate the use of the same.
9. Write a program in Java to convert an image in RGB to a Grayscale image.
10. Develop Chat Server using Java.

Outcomes:

At the end of the course, the student should be able to:

- Develop error-free and well-documented Java programs.
- Develop and test Java network, search engine, and web framework programs.
- Learn how to write, test, and debug advanced-level Object-Oriented programs using Java.

RDBMS (ORACLE) LAB

Objectives: The major objective of this lab is to provide a strong formal foundation in database concepts, its implementations, techniques relating to query processing by SQL Engine

1. Creating & updating and inserting into database & simple queries.

2. Uses of Select statement - for queries.

a. AND' OR' NOT Operators' WHERE clause.
b. UNION' INTERSECTION' MINUS.
c. Sorting and grouping.

3. Nested queries using SQL. a. Sub queries. b. Join.

4. Built-in-functions of SQL.

5. Use of indexes' creating views and querying in views.

6. Cursors' triggers and stored procedures and functions.

7. **Case Studies:**

a. Student evaluation systems.
b. Pay - roll system
c. Income tax calculations.
d. Seat reservation Problems
e. Mark - sheet Preparation.

Outcomes:

After undergoing this laboratory module, the participant should be able to:

- Understand, appreciate and effectively explain the underlying concepts of database technologies
- Design and implement a database schema for a given problem-domain
- Normalize a database
- Populate and query a database using SQL DML/DDL commands.
- Declare and enforce integrity constraints on a database using a state-of-the-art RDBMS
- Programming PL/SQL including stored procedures, stored functions, cursors, packages.

****

OPEN SOURCE TECHNOLOGIES (PHP AND MYSQL)

OBJECTIVES: The objective is to teach the students the principles of open source technology, benefits of open source, and the product that includes permission to use its source code, design documents, or contents. The students will also study and understand the different open source licenses and how to start an open source project.

UNIT I: OPEN SOURCE : Introduction : Open Source – Open Source vs. Commercial Software – What is Linux? - Free Software – Where I can use Linux? Linux Kernel – Linux Distributions

UNIT II: LINUX : Introduction: Linux Essential Commands - Filesystem Concept - Standard Files - The Linux Security Model - Vi Editor - Partitions creation - Shell Introduction - String Processing - Investigating and Managing Processes - Network Clients - Installing Application

UNIT III: APACHE : Introduction - Apache Explained - Starting, Stopping, and Restarting Apache - Modifying the Default Configuration - Securing Apache - Set User and Group - Consider Allowing Access to Local Documentation - Don't Allow public_html Web sites - Apache control with .htaccess

UNIT IV: MySQL : Introduction to MY SQL - The Show Databases and Table - The USE command - Create Database and Tables - Describe Table - Select, Insert, Update, and Delete statement - Some Administrative detail - Table Joins - Loading and Dumping a Database.

UNIT V: PHP : PHP Introduction- General Syntactic Characteristics - PHP Scripting - Commenting your code - Primitives, Operations and Expressions - PHP Variables - Operations and Expressions Control Statement - Array - Functions - Basic Form Processing - File and Folder Access - Cookies - Sessions - Database Access with PHP - MySQL - MySQL Functions - Inserting Records - Selecting Records - Deleting Records - Update Records.

Course Outcomes:

On successful completion of this course students should be able

1. To develop android applications.
2. To install and work on Linux.
3. To perform Shell Programming.

TEXT BOOK

1. "Open Source Web Development with LAMP using Linux, Apache, MySQL, Perl and PHP", James Lee and Brent Ware, Dorling Kindersley(India) Pvt. Ltd, 2008

REFERENCE BOOKS

1. "Setting Up LAMP: Getting Linux, Apache, MySQL, and PHP and working Together", Eric Rosebrock, Eric Filson, Published by John Wiley and Sons, 2004

PRINCIPLES OF COMPILER DESIGN

Objective: An introduction to the design and implementation of programming language translators. Theoretical aspects of language design and translation are discussed and practically demonstrated by developing a working compiler.

Unit I

Introduction on the phase of the complier – Lexical Analysis, Regular Expression, Non deterministic Automata, Deterministic Automata equivalent to NFA's. Minimizing the states of DFA, Implementation of Lexical Analyzer.

Unit II

Syntax Analysis – Context free grammars - Top down Parsing Concepts, Recursive Descent Parsing, Predictive Parsers, Non recursive Pedictive Parsing – Bottom Up Parsing, Handle pruning, Shift reduce parsing – Operator Predence Parsing – Error recovery in Parsing, Parser Generators – YACC. [Omit LR Parsers : Sec 4.7]

Unit III

Intermediate Code Generation: Syntax directed Definitions, Construction of Syntax trees – Top down Translation, Bottom up Evaluation of inherited Attributes, Recursive Evaluators, Assigning Space at Complier Construction time – Type checking.

Unit IV

Storage Organization : Storage Organization, Storage Allocation Strategies, Parameter Passing, Symbol tables, Dynamic Storage Allocation, Intermediate Languages – Representation of Declarations, Assignment Statement, Boolean Expression, Back patching, Procedure calls.

Unit V

Code Generation and Optimization: Design of the code generators, Runtime storage Management, Basic blocks and flow graphs, Register Allocation and Assignment, DAG representation of Basic blocks, Peephole optimization, Code optimization – The principle sources of optimization, Optimization of basic blocks, Global data flow Analysis, Loop optimizations.

OUTCOMES:

On successful completion of the course students will be able to:

1. Specify and analyse the lexical, syntactic and semantic structures of advanced language features

2. Separate the lexical, syntactic and semantic analysis into meaningful phases for a compiler to undertake language translation

3. Write a scanner, parser, and semantic analyser without the aid of automatic generators

4. Turn fully processed source code for a novel language into machine code for a novel computer

5. Describe techniques for intermediate code and machine code optimisation

6. Design the structures and support required for compiling advanced language features.

Text Book(s)

1. Alfred V.Aho, Ravi Sethi, Jeffrey D.Ullman, "Compilers – Principles, Techniques and Tools", 2007, Pearson Education.

2. Dhamdhere D.M., "Compiler Construction Principles and Practice", 1981, Macmillan India.

References

1. Reinhard Wilhlm, Director Mauser, "Compiler Design", 1995, Addison Wesley.

.NET FRAMEWORK

OBJECTIVE: Understand the concept of .NET framework, study the different techniques of security, introduce web services with ASP.NET, and explore window based applications.

UNIT I INTRODUCTION TO C#

Introducing C#, Understanding .NET, overview of C#, Literals, Variables, Data Types, Operators, checked and unchecked operators, Expressions, Branching, Looping, Methods, implicit and explicit casting, Constant, Arrays, Array Class, Array List, String, String Builder, Structure, Enumerations, boxing and unboxing.

UNIT II OBJECT ORIENTED ASPECTS OF C#

Class, Objects, Constructors and its types, inheritance, properties, indexers, index overloading, polymorphism, sealed class and methods, interface, abstract class, abstract and interface, operator overloading, delegates, events, errors and exception, Threading.

UNIT III APPLICATION DEVELOPMENT ON .NET

Building windows application, Creating our own window forms with events and controls, menu creation, inheriting window forms, SDI and MDI application, Dialog Box(Modal and Modeless), accessing data with ADO.NET, DataSet, typed dataset, Data Adapter, updating database using stored procedures, SQL Server with ADO.NET, handling exceptions, validating controls, windows application configuration.

UNIT IV WEB BASED APPLICATION DEVELOPMENT ON .NET

Programming web application with web forms, ASP.NET introduction, working with XML and .NET, Creating Virtual Directory and Web Application, session management techniques, web.config, web services, passing datasets, returning datasets from web services, handling transaction, handling exceptions, returning exceptions from SQL Server.

UNIT V CLR AND .NET FRAMEWORK

Assemblies, Versoning, Attributes, reflection, viewing meta data, type discovery, reflection on type, marshalling, remoting, security in .NET

OUTCOMES:

After completing this course, the student will be able to

List the major elements of the .NET frame work
Explain how C# fits into the .NET platform.
Analyze the basic structure of a C# application
Debug, compile, and run a simple application.
Develop programs using C# on .NET
Design and develop Web based applications on .NET
Discuss CLR.

TEXT BOOKS:

1. Herbert Schildt, “The Complete Reference: C# 4.0”, Tata McGraw Hill, 2012.
2. Christian Nagel et al. “Professional C# 2012 with .NET 4.5”, Wiley India, 2012.

REFERENCES:

1. Andrew Troelsen , “Pro C# 2010 and the .NET 4 Platform, Fifth edition, A Press, 2010.
2. Ian Griffiths, Matthew Adams, Jesse Liberty, “Programming C# 4.0”, Sixth Edition, O‟Reilly, 2010.

EMERGING TRENDS AND TECHNOLOGY

Subject Code:	Credits:	External Marks: 75	Hours:

OBJECTIVE: As technology plays a larger role in education, any predictions concerning the future of education must include an analysis of technological trends.

UNIT I: **Trends in Computing**: Introduction to Parallel Computing –Scope of Parallel Computing – parallel programming platforms: Implicit parallelism trend in microprocessor architecture – Limitations of memory system performances - Introduction to mobile computing – Applications – vehicles- Mobile and Wireless devices- History of wireless communications- A Simplified and reference model – Multiplexing – Modulation - Cloud computing –Cloud Architecture – Cloud Storage –Advantages and Disadvantages of Cloud computing –Cloud Services .

UNIT-II: **Introduction to Internet of Things:** Introduction, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels & Deployment Templates, Domain Specific IoTs: Home Automation, Cities, Environment, Energy, Retail, Agriculture, Health & Lifestyle.

UNIT-III: **Portable Devices Programming:** IoT Physical Devices & Endpoints: What is an IoT Device, Exemplary Device: Raspberry Pi, About the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces, programming Raspberry Pi with Python, Other IoTDevices.

UNIT-IV: **Block Chain Technology:** Overview of Blockchain Technology – What is Blockchain – Transactions – Blocks – Hashes – Concensus – Verify and Confirm Blocks – Hashes: Hash Cryptography – Encryption Vs Hashing – Transactions: Recording Transactions – Digital Signature – Verifying and Confirming Transactions. Blocks and Blockchain: Hash Pointers – Blocks – Concensus Building – Distributed Concensus – Byzantine Generals Problem – Concensus Mechanism – Blockchain Architecture – Markle Root Tree – Blockchain and future world of Web 3.0.

UNIT-V: Introduction to Virtual Reality (VR), modern experiences, historical perspective, Needs of VR, Bird's-Eye View, Hardware, sensors, displays, software, virtual

world generator, game engines, human senses, perceptual psychology, psychophysics. Augmented Reality (AR): Introduction and brief history of AR, AR today, Difference between AR and VR.Challenges for AR, opportunities for augmented reality. Types of augmented reality: AR function, AR methods, AR Display technology.

COURSE OUTCOMES:

- Discuss the impact of disruptive technologies on project design, implementation, and transformation.
- Identify major areas where technologies can be applied and their implications for organizational change.
- Recognize current and emerging disruptive technologies and their potential to impact social conditions, the economy, and daily life.
- Design a project plan that incorporates a new and emerging technology and illustrates its impact on organizations and industries.

REFERENCE BOOKS:

1. Introduction to Parallel Computing, 2nd Edition, by Anshul Gupta, AnanthGrama, George Karypis, Vipinkumar, 2003, Publisher Addision Wesley.
2. Mobile Communication, JochenSchiler, AddisionWelsey, 2000.
3. Cloud Computing Concepts, Technology and Architecture by Thomas Ere- May2013.
4. Virtual Reality, Steven M. LaValle,University of Illinoise, Cammbridge University Press.
5. Augmented Reality, Greg Kipper, Joseph Rampolia, Elsevier Science.
6. ArshdeepBahga, Vijay Madisetti, Internet of Things – A Hands on Approach, University Press.

NEURAL NETWORK AND FUZZY SYSTEM

OBJECTIVES:

To master the various fundamental concepts of fuzzy logic and artificial neural networks. This will help you to get sufficient knowledge to analyze and design the various intelligent control systems

UNIT I FUNDAMENTALS OF FUZZY LOGIC

Basic concepts: fuzzy set theory- basic concept of crisp sets and fuzzy sets-complements- union- intersection- combination of operation- general aggregation operations- fuzzy relations-compatibility relations-orderings- morphisms- fuzzy relational equations-fuzzy set and systems

UNIT II ARCHITECTURE OF NEURAL NETWORKS

Architectures: motivation for the development of natural networks-artificial neural networks-biological neural networks-area of applications-typical Architecture-setting weights-common activations functions- Basic learning rules- Mcculloch-Pitts neuron-Architecture, algorithm, applications-single layer net for pattern classification- Biases and thresholds, linear separability - Hebb'srule- algorithm -perceptron - Convergence theorem-Delta rule

UNIT III BASIC NEURAL NETWORK TECHNIQUES

Back propagation neural net:standard back propagation-architecture algorithm-derivation of learning rules- number of hidden layers--associative and other neural networks- hetro associative memory neural net, auto associative net- Bidirectional associative memory-applications-Hopfield nets-Boltzman machine

UNIT IV COMPETITIVE NEURAL NETWORKS

Neural network based on competition: fixed weight competitive nets- Kohonenself organizing maps and applications-learning vector quantization-counter propagation nets and applications adaptive resonance theory: basic architecture and operation-architecture, algorithm, application and analysis of ART1 & ART2

UNIT V SPECIAL NEURAL NETWORKS

Cognitron and Neocognitron - Architecture, training algorithm and application- fuzzy associate memories, fuzzy system architecture- comparison of fuzzy and neural systems.

OUTCOMES:

1. To understand the basic concept of fuzzy sets, fuzzy logic & defuzzification
2. To learn basics of Artificial Neural of theory and programming of Microprocessors
3. To analyze various techniques in feedback and feed forward Neural networks.
4. To Understand the principle of competitive neural networks and Adaptive resonance theory
5. To learn the architecture and algorithm of Cognitron, Neo cognitron The concepts of fuzzy associative memory and fuzzy systems.

Text book(s) and/or required material

1. T1. Kliryvan- Fuzzy System & Fuzzy logic Prentice Hall of India, First Edition.
2. Lawrence Fussett- fundamental of Neural network Prentice Hall , First Edition.

Reference Books:

- Bart Kosko, –Neural network and Fuzzy System - Prentice Hall-1994.
- J.Klin and T.A.Folger, –Fuzzy sets University and information- Prentice Hall -1996.
- J.M.Zurada, –Introduction to artificial neural systems -Jaico Publication house,Delhi 1994.
- VallusuRao and HayagvnaRao , –C++ Neural network and fuzzy logic -BPB and Publication, New Delhi,1996.

OPEN SOURCE TECHNOLOGY LAB

Objectives: To provide fundamental concept of internet, with a view to developing professional software development skills. To expose students to free open source software environment and introduce them to use open source packages.

1. Write a server side php program that display marks, total, grade of student in tabular format by accepting user inputs for name, number and marks from a HTML form.
2. Write a php program that adds products that a selected from a web page to a shopping cart.
3. Write a php program to access the data stored in my sql table.
4. Write a php program interface to create a database and insert a table
   a) Write a php program using classes to create a table
5. Write a php program to create a directory and read contents from the directory
6. Write a shell program to find the details of an user session
7. Write ashell program to change the extension fo a give file
8. Create a mysql table and execute queries to read, add, remove and modify a record from the table.
9. Create a php program for USER AUTHENTICATION
10. Create a php program for UPLOAD FILE TO SERVER

Course Outcomes: At the end of this course student will:

CO1) Implement various applications using build systems

CO2) Understand the installation of various packages in open source operating systems

CO3) Create simple GUI applications using PHP

CO4) Understand various version control sytems

.NET FRAMEWORK LAB

OBJECTIVES:

This Lab course will help students to achieve the following objectives:

1.Introduce to .Net IDE Component Framework.

2. Programming concepts in .Net Framework.

3. Creating website using ASP.Net Controls.

LIST:-

1. Create a windows form with the following controls Textbox, Radio button, Check box, Command Button
2. Write a program for Menu option.
3. Create a program to connect with database and manipulate the records in the database using ADO .NET
4. Create a program to implement the concepts of OOPS for creating class, inheritance
5. Create a program to perform input validation using procedure.
6. Write a program to open a file and using I/O operations write contents into a file and read the contents from the file.
7. Create a window form using HTML controls.
8. Create a program to perform validation using validation controls.
9. Create a program in ASP .NET to connect with the database using ADODB connectivity and manipulate the records.
10. Write a program to store the employee details using class and methods in C# .NET
11. Write a program to Handle Exceptions
12. Write a program to create a form with Basic controls. In c#. NET.

OUTCOMES:

At the end of this Lab course students will be able to:

2.Create user interactive web pages using ASP.Net.

3.Create simple data binding applications using ADO.Net connectivity.

4. Performing Database operations for Windows Form and web applications.

SOFTWARE ENGINEERING

OBJECTIVES:

- To understand Software Engineering Lifecycle Models
- To do project management and cost estimation
- To gain knowledge of the System Analysis and Design concepts.
- To understand software testing approaches
- To be familiar with DevOps practices

UNIT 1: **Software Engineering :** Definition – Layers – Management – S/W Engineer – Phases in S/W Engineering – Software Process, Project and Product – S/W Life Cycle Models : Waterfall Model , Prototyping Model, Incremental Model, Spiral Models – S/W Requirements : Types of Requirements – Requirements Engineering Process – Requirements Elicitation – Requirements Validation.

UNIT 2: **Software Design:** Basics of Software Design – Architectural Design – Object-Oriented Design – Software Design Notations – Software Design Reviews – Software Coding: Features of Software Code – Coding Methodology – Programming Practice – Code verification techniques - Coding Tools.

UNIT 3 : **Software Testing:** Software Testing Basics – Software Testing Strategies – Levels of Software Testing: Unit Testing – Integration Testing – System Testing – Acceptance Testing – Testing Techniques : White Box Testing –Black Box Testing – Software Testing Tools- Debugging process – Debugging strategies.

UNIT 4 : **Software Maintenance :** Basics of S/W Maintenance – Types of S/W Maintenance. S/W Planning and Scheduling: Responsibilities of S/W Project Manager – Project Planning – Project Scheduling – Project Staffing – S/W Maintanence models - Techniques for maintenance – S/W Maintenance tools.

UNIT 5 : **Software Cost Estimation :** Basics of Cost Estimation – S/W Cost Estimation Process – Software Estimation Models: COCOMO Model – Software Equation – Expert Judgment. Software Quality : Quality Concepts – S/W Quality Assurance Group – Evaluation of Quality – Advanced Topics: Client/Server S/w Engineering – Web Engineering.

OUTCOMES:

At the end of this course, the students will be able to:

- Understand the advantages of various Software Development Lifecycle Models
- Gain knowledge on project management approaches as well as cost and schedule estimation strategies
- Perform formal analysis on specifications
- Architect and design using architectural styles and design patterns

- Understand software testing approaches
- Understand the advantages of DevOps practices

TEXT BOOK:

1.” Software Engineering” Rohit Khurana ,ITL ESL, vikas II Edition,2014.

REFERENCE BOOKS:

1.”Software Engineering – An Engineering Approach”, James F Peters And Witold Pedryez, John Wiley and Sons, New Delhi,2000.

2.”Software Engineering – A Practitioner’s Approach”, Roger S.Pressman, Mcgraw-Hill International Edittion,6th Edition,2005.

DATA SCIENCE

Objectives : After the completion of the Data Science course, you should be able to:
1. Gain insight into the 'Roles' played by a Data Scientist
2. Analyse Big Data using R, Hadoop and Machine Learning
3. Understand the Data Analysis Life Cycle
4. Work with different data formats like XML, CSV and SAS, SPSS, etc.
5. Learn tools and techniques for data transformation
6. Understand Data Mining techniques and their implementation
7. Analyse data using machine learning algorithms.

UNIT I: Data Evolution: Data Development Time Line – Data Growth - a Perspective – IT Components - Business Process –Data to Data Science. **Understanding data**: Introduction – Type of Data: Numeric – Categorical – Graphical – High Dimensional Data — Data Classification – Hot Data – Cold Data – Warm Data – Thick Data – Thin Data - Classification of digital Data: Structured, Semi-Structured and Un-Structured. Sources of Data: Time Series – Transactional Data – Biological Data – Spatial Data – Social Network Data – Data Evolution – Data Sources.

UNIT II: Data Science: Data Science-A Discipline – Data Science vs Statistics, Data Science vs Mathematics, Data Science vs Programming Language, Data Science vs Database, Data Science vs Machine Learning – Introduction to machine learning algorithm.

UNIT III : ADVANCED ANALYTICAL THEORY AND METHODS - Overview of Clustering, K-means, Use Cases, Overview of the Method, Perform a K-means Analysis using R, Classification, Decision Trees, Overview of a Decision Tree, Decision Tree Algorithms, Decision Tree in R, Bayes‘ Theorem Evaluating a Decision Tree, Decision Tree in R, Bayes‘ Theorem, Hypothesis Testing, Difference of Means, Wilcoxon Rank-Sum Test, ANOVA.

UNIT IV: Big Data: Digital Data-an Imprint: Evolution of Big Data – What is Big Data – Sources of Big Data. Characteristics of Big Data 6Vs – Big Data Myths - Data Discovery-Traditional Approach, Big Data Technology: Big Data Technology Process – Big Data Exploration - Data Augmentation – Operational Analysis – 360 View of Customers – Security and Intelligence

UNIT V: ANALYTICS TECHNOLOGY AND TOOLS - Analytics for Unstructured Data, Use Cases, MapReduce, Apache Hadoop, The Hadoop Ecosystem **. HADOOP DISTRIBUTED FILE SYSTEM ARCHITECTURE-** HDFS Architecture, HDFS Concepts, Blocks, NameNode, Secondary NameNode, DataNode, HDFS Federation, HDFS High Availability, Basic File System Operations, Data Flow, Anatomy of File Read, Anatomy of File Write, Anatomy of a MapReduce Job Run- **INTRODUCTION TO APACHE SPARK FRAME WORK** – overview of spark API -Scala, Python.

COURSE OUTCOMES:

- The students will be able to:
- Identify Big Data and its Business Implications.
- List the components of Hadoop and Hadoop Eco-System
- Access and Process Data on Distributed File System
- Manage Job Execution in Hadoop Environment
- Develop Big Data Solutions using Hadoop Eco System
- Analyze Infosphere BigInsights Big Data Recommendations.
- Gain knowledge in Machine Learning Techniques

TEXT BOOK:

1. David Dietrich, Barry Heller and Beibei Yang, ―Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data‖ , EMC Education Services, Reprint 2015, Wiley, ISBN: 9788126556533
2. V. Bhuvaneswari, T. Devi, "Big Data Analytics: A Practitioner's Approach" 2016.

Reference Books

1. Han Hu, Yonggang Wen, Tat-Seng, Chua, Xuelong Li,"Toward Scalable Systems for Big Data Analytics: A Technology Tutorial", IEEE, 2014

2. Tom White, "Hadoop: The Definitive Guide", 4th Edition, 2015, O'Reilly, ISBN: 9789352130672.

3. BirisLublinsky, Kevin T. Smith and Alexey Yakubovich, ―Professional Hadoop Solutions‖ , Reprint 2014, Wiley, ISBN 13:9788126551071.

4. Stephen Marsland, ―Machine Learning – An Algorithmic Perspective‖ , , Taylor& Francis Group, Second Edition, 2015, Chapman & Hall / CRC Press , ISBN:9781466583283.

5. Nathan Marz, James Warren, ―Big Data-Principles and best practices of scalable real-time data systems‖ , Edition 2015, DreamTech Press, ISBN: 9789351198062

PYTHON PROGRAMMING

OBJECTIVE: To Obtain Knowledge from the Python programming and Data Analytics Using Python.

UNIT I: Introduction to Python Programming: Arithmetic operator- values and types. variables, expressions and statements: Assignment Statement - Variable Names – Expressions and Statement –String Operation. Function: Function Calls – Math Function - Adding new Functions - Definitions and Uses - Flow of execution, parameters and arguments -Fruitful functions.

UNIT II: Conditionals and Recursion: Floor Division and Modulus - Boolean Expression – Logical operators – Conditional Execution- Alternative Executive- chained conditional - Nested conditionals – Iteration – Strings. Lists, Tuples, Dictionaries: Lists: A List is a Sequence – List are Mutable - Traversing a List - List operations - List slices - List methods - Map, Filter and Reduce – Deleting Elements – List and String – List Arguments – Dictionaries –Tuples – Files – Introduction to packages :Package types – Package Installation.

UNIT III: Introduction to Data Analysis: Data Analysis – Knowledge domain of the data Analysist – The Data Analysis Process – Quantitative and Qualitative Data Analysis –Scipy. Numpy Library: Basic Operations – Indexing,Slicing,Iterating – Array Manipulatrion.

UNIT IV: Pandas and Data Visualization: Pandas Data Structure-Series – The Data Frame – The Index Object- Sorting and Ranking – Correlation and Covariance .Pandas Reading and Writing: Csv, Text File, HTML Files, JSON Data-Interacting With database – Matplotlib Architecture – pyplot- Line Chart – Bar Chart –Pie Chart.

UNIT V: Machine Learning With SciKit: Supervised and Unsupervised Learning – Training Set and Testing set - Supervised Learning with scikit Learn The Iris Flower Dataset – K–Nearest Neighbors Classifier – Diabetes Dataset – Linear Regression:The Least Square Regression – Support Vector Machine(SVMs).

OUT COMES:

After completing this subject the student will be able to

- Gain programming Knowledge to solve Data science task through this language.
- Gain programming Knowledge to solve Data Analytics task through this Language.
- Gain programming Knowledge to solve Big Data Analytics task through this Language.

TEXT BOOK:

Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist‘‘, 2nd edition, Updated for Python 3, Shroff/O‘Reilly Publishers, 2016.

“Python Data Analytics”, Fabio Nelli, Pub: APress

ARTIFICIAL INTELLIGENCE & EXPERT SYSTEMS

Objective:

Be able to implement a wide range of AI concepts using Prolog.
Be able to assess the potential of AI in research and real-world environments.

UNIT I: The AI Problems: AI Techniques – Problems, problems space and search – Defining the problem as a state space search – Production systems – Problem Characteristics – Production system characteristics – Heuristic search techniques – Generate and test – Hill climbing – Best first search – Problem reduction – Constraint satisfaction – Means end analysis.

UNIT II: Game playing – Mini max search procedure – Adding alpha beta cutoffs– Additional Refinement – Iterative deepening – Using predicate logic – Representing simple facts and Logic – Representing Instance and Isa relationships – Computable functions and predicates – Resolution – Natural deduction.

UNIT III: Representing knowledge using rules – Procedural verses declarative knowledge – Logic programming – forward verses backward reasoning – matching – control knowledge – symbolic reasoning under uncertainty – non monotonic reasoning – Logic for non monotonic reasoning – Implementation issues – Augmenting a problem solver – Implementation of depth first search – Implementation of breath first search.

UNIT IV: Statistical reasoning – Probability and Bayee's theorem – Certainty factors and Rule based systems – Bayesian Network – Dempster – Shafer theory – Fuzzy logic. Parallel and Distributed AI: Psychological Modeling – Parallelism in Reasoning systems – Distributed Reasoning systems.

UNIT V: Expert System – Features of an Expert System – Persons Involved in Expert System Building Organizing Knowledge – Representing Knowledge – Frame based Methods – Basic characteristics of an Expert System – Expert Systems make mistakes – Knowledge Representation using Rules – Knowledge Representation using Semantic Nets – Knowledge Representation using Frames.

Outcomes: At the end of the course, the student should be able to:

- Gain a working knowledge of the foundations of, and modern applications in, artificial intelligence, including agent design, heuristic search, knowledge representation, planning, logic, natural language processing and machine learning.
- Gain experience writing several AI applications in a variety of programming languages.

Text Books:

Artificial Intelligence, Elaine Rich and Kevin Knight, Tata McGraw Hill, 2nd Edition, 1991.
A Guide to Expert Systems, Donald A. Waterman, Pearson Education.

E – Commerce

Objectives: This course provides an introduction to information systems for business and management. It is designed to familiarize students with organizational and managerial foundations of systems, the technical foundation for understanding information systems

UNIT I: INTRODUCTION

Networks and Commercial Transactions - Internet and Other Novelties - Electronic Transactions Today - Commercial Transactions - Establishing Trust - Internet Environment - Internet Advantage - World Wide Web.

UNIT II: SECURITY TECHNOLOGIES

Why Internet Is Unsecure - Internet Security Holes - Cryptography : Objective - Codes and Ciphers - Breaking Encryption Schemes - Data Encryption Standard - Trusted Key Distribution and Verification - Cryptographic Applications - Encryption - Digital Signature - Nonrepudiation and Message Integrity.

UNIT III: ELECTRONIC PAYMENT METHODS

Traditional Transactions : Updating - Offline and Online Transactions - Secure Web Servers - Required Facilities - Digital Currencies and Payment Systems - Protocols for the Public Transport - Security Protocols - SET - Credit Card Business Basics.

UNIT IV: ELECTRONIC COMMERCE PROVIDERS

Online Commerce Options - Functions and Features - Payment Systems : Electronic, Digital and Virtual Internet Payment System - Account Setup and Costs - Virtual Transaction Process - InfoHaus - Security Considerations – CyberCash: Model - Security - Customer Protection - Client Application - Selling through CyberCash.

UNIT V: ONLINE COMMERCE ENVIRONMENTS

Servers and Commercial Environments - Payment Methods - Server Market Orientation - Netscape Commerce Server - Microsoft Internet Servers - Digital Currencies - DigiCash - Using Ecash - Ecash Client Software and Implementation - Smart Cards - The Chip - Electronic Data Interchange - Internet Strategies, Techniques and Tools.

Outcomes:

- After Completion of the subject student should able to
- Understand the basic concepts and technologies used in the field of management information systems;
- Have the knowledge of the different types of management information systems;
- Understand the processes of developing and implementing information systems;
- Be aware of the ethical, social, and security issues of information systems;

TEXT BOOKS

1.Pete Loshin, "Electronic Commerce", 4th Edition, Firewall Media, An imprint of Laxmi Publications Pvt. Ltd., New Delhi, 2004.

REFERENCES

1. Jeffrey F.Rayport and Bernard J. Jaworski, "Introduction to E-Commerce", 2nd Edition, Tata Mc-Graw Hill Pvt., Ltd., 2003.
2. Greenstein, "Electronic Commerce", Tata Mc-Graw Hill Pvt., Ltd., 2000.

Human Computer Interaction

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

- Design, implement and evaluate effective and usable graphical computer interfaces.
- Describe and apply core theories, models and methodologies from the field of HCI.
- Describe and discuss current research in the field of HCI.
- Implement simple graphical user interfaces using the Java Swing toolkit.
- Describe special considerations in designing user interfaces for older adults.

Unit I

The interaction: Introduction – Models of interaction – Frameworks and HCI – Ergonomics – Interaction Styles – Elements of WIMP interface – Interactivity – The Context of the interaction - **Paradigm**: Introduction – Paradigms for interaction.

Unit II

Interaction Design basics: Introduction – what is design? – User focus – Scenarios – Navigation design – Screen design and layout – Interaction and prototyping - **HCI in the software process:** Introduction **–** The software lifecycle – Usability engineering – Interactive design and prototyping – Design rationale.

Unit III

Design rules: Introduction – Principles to support usability – Standards – Guidelines – Golden rules and heuristics – HCI patterns - **ImplementationSupport**: Introduction – Elements of windowing systems – Programming the application – Using toolkits – User interface management systems.

Unit IV

Evaluation techniques: What is evaluation – Goals of evaluation – Evaluation through expert analysis – Evaluation through user participation – Choosing an evaluation method - **Universal Design**: Introduction – Universal design principles – Multi-modal interaction – Designing for diversity.

Unit V

User Support: Instruction – Requirements of user support – Approaches to user support – Adaptive help system – Designing user support systems.

Outcomes

Upon completion of the course, students should be able to:

- Explain the capabilities of both humans and computers from the viewpoint of human information processing.
- Describe typical human–computer interaction (HCI) models and styles, as well as various historic HCI paradigms.
- Apply an interactive design process and universal design principles to designing HCI systems.
- Describe and use HCI design principles, standards and guidelines.
- Analyze and identify user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems.
- Discuss tasks and dialogs of relevant HCI systems based on task analysis and dialog design.
- Analyze and discuss HCI issues in groupware, ubiquitous computing, virtual reality, multimedia, and Word Wide Web-related environments.

Text Book: 1. "Human-computer Interaction" - Alan Dix - Pearson Education - 2004.

CLOUD COMPUTING

OBJECTIVES:

To understand the concept of cloud computing.

To appreciate the evolution of cloud from the existing technologies.

To have knowledge on the various issues in cloud computing.

To be familiar with the lead players in cloud.

To appreciate the emergence of cloud as the next generation computing paradigm.

Unit I

Introduction: History of Centralized and Distributed Computing – Overview of Distributed Computing, Cluster computing, Grid computing. Technologies for Network-based systems- System models for Distributed and cloud computing- Software environments for distributed systems and clouds.

Unit II

Virtualization: Introduction to Cloud Computing- Cloud issues and challenges – Properties – Characteristics – Service models, Deployment models. Cloud resources: Network and API – Virtual and Physical computational resources – Data-storage. Virtualization concepts – Types of Virtualization- Introduction to Various Hypervisors – High Availability (HA)/Disaster Recovery (DR) using Virtualization, Moving VMs.

Unit III

Service Models: Infrastructure as a Service (IaaS) – Resource Virtualization: Server, Storage, Network – Case studies. Platform as a Service (PaaS) – Cloud platform & Management: Computation, Storage – Case studies. Software as a Service (SaaS) – Web services – Web 2.0 – Web OS – Case studies – Anything as a service (XaaS) – Microservices.

Unit IV

Cloud Programming and Software Environments: Cloud Programming and Software Environments – Parallel and Distributed Programming paradigms – Current technologies – Programming support of App Engines – Emerging Cloud software Environment.

Unit V

Cloud Access: authentication, authorization and accounting – Cloud Provenance and meta-data – Cloud Reliability and fault-tolerance – Cloud Security, privacy, policy and compliance- Cloud federation, interoperability and standards.

Course Outcomes

a. Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure. b. Compare the advantages and disadvantages of various cloud computing platforms.

c. Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.

d. Program data intensive parallel applications in the cloud.

REFERENCE

1. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, "Distributed and cloud computing from Parallel Processing to the Internet of Things", Morgan Kaufmann, Elsevier, 2012.
2. Barrie Sosinsky, "Cloud Computing Bible", John Wiley & Sons, 2010. 2. Tim Mather, Subra Kumaraswamy, and Shahed Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly, 2009.

PARALLEL COMPUTING

OBJECTIVES:

- To familiarize the issues in parallel computing.
- To describe distributed memory programming using MPI.
- To understand shared memory paradigm with Pthreads and with OpenMP.
- To learn the GPU based parallel programming using OpenCL.

UNIT I FOUNDATIONS OF PARALLEL PROGRAMMING

Motivation for parallel programming – Need-Concurrency in computing – Basics of processes, multitasking and threads – cache – cache mappings – caches and programs – virtual memory – Instruction level parallelism – hardware multi-threading – Parallel Hardware-SIMD – MIMD – Interconnection networks – cache coherence –Issues in shared memory model and distributed memory model –Parallel Software- Caveats- coordinating processes/ threads- hybrid model – shared memory model and distributed memory model - I/O – performance of parallel programs-– parallel program design.

UNIT II DISTRIBUTED MEMORY PROGRAMMING WITH MPI

Basic MPI programming – MPI_Init and MPI_Finalize – MPI communicators – SPMD-programs– MPI_Send and MPI_Recv – message matching – MPI- I/O – parallel I/O – collective communication – Tree-structured communication -MPI_Reduce – MPI_Allreduce, broadcast, scatter, gather, allgather – MPI derived types – dynamic process management – performance evaluation of MPI programs- A Parallel Sorting Algorithm

UNIT III SHARED MEMORY PARADIGM WITH PTHREADS

Basics of threads, Pthreads – thread synchronization – critical sections – busy waiting – mutex – semaphores – barriers and condition variables – read write locks with examples - Caches, cache coherence and false sharing – Thread safety-Pthreads case study.

UNIT IV SHARED MEMORY PARADIGM: OPENMP

Basics OpenMP – Trapezoidal Rule-scope of variables – reduction clause – parallel for directive – loops in OpenMP – scheduling loops –Producer Consumer problem – cache issues – threads safety in OpenMP – Two- body solvers- Tree Search

UNIT V GRAPHICAL PROCESSING PARADIGMS: OPENCL AND INTRODUCTION TO CUDA

Introduction to OpenCL – Example-OpenCL Platforms- Devices-Contexts - OpenCL programming – Built-In Functions-Programs Object and Kernel Object – Memory Objects - Buffers and Images – Event model – Command-Queue - Event Object - case study. Introduction to CUDA programming.

OUTCOMES:

- Identify issues in parallel programming.
- Develop distributed memory programs using MPI framework.
- Design and develop shared memory parallel programs using threads and using OpenMP.
- Implement Graphical Processing OpenCL programs.

REFERENCES:

1. A. Munshi, B. Gaster, T. G. Mattson, J. Fung, and D. Ginsburg, ―OpenCL programming guide‖, Addison Wesley, 2011
2. M. J. Quinn, ―Parallel programming in C with MPI and OpenMP‖, Tata McGraw Hill, 2003.
3. Peter S. Pacheco, ―An introduction to parallel programming‖, Morgan Kaufmann, 2011.
4. Rob Farber, ―CUDA application design and development‖, Morgan Haufmann, 2011.
5. W. Gropp, E. Lusk, and A. Skjellum, ―Using MPI: Portable parallel programming with the message passing interface‖, Second Edition, MIT Press, 1999

GRID COMPUTING

Objective:

To provide a thorough knowledge about the technology application and tool kits for grid computing.

UNIT I CONCEPTS AND ARCHITECTURE

Introduction-Parallel and Distributed Computing-Cluster Computing-Grid Computing-Anatomy and Physiology of Grid- Web and Grid Services-Grid Standards - OGSA-WSRF - Trends, Challenges and applications.

UNIT II GRID MONITORING

Grid Monitoring Architecture (GMA) - An Overview of Grid Monitoring Systems- R-GMA - GridICE – MDS- Service Level Agreements (SLAs) - Other Monitoring Systems- Ganglia, GridMon, Hawkeye and Network Weather Service.

UNIT III GRID SECURITY AND RESOURCE MANAGEMENT

Grid Security-A Brief Security Primer-PKI-X509 Certificates-Grid Security-Grid Scheduling and Resource Management, Gridway and Gridbus Broker-principles of Local Schedulers-Overview of Condor, SGE, PBS, LSF-Grid Scheduling with QoS.

UNIT IV DATA MANAGEMENT AND GRID PORTALS

Data Management-Categories and Origins of Structured Data-Data Management Challenges- Architectural Approaches-Collective Data Management Services-Federation Services-Grid Portals-Generations of Grid Portals.

UNIT V GRID MIDDLEWARE

List of globally available Middlewares - Case Studies-Recent version of Globus Toolkit and gLite - Architecture, Components and Features. Features of Next generation grid.

Course Outcomes :

- Understanding the fundamentals of grid computing.
- Discussing the basics of grid monitoring.
- Learning the concepts of grid security and resource management.
- Understanding the concepts of grid portals.
- Understanding the advanced grid middleware.

REFERENCES

1. Ian Foster, Carl Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure, Elsevier Series, 2004.
2. Vladimir Silva, Grid Computing for Developers, Charles River Media, January 2006.
3. Parvin Asadzadeh, RajkumarBuyya, Chun Ling Kei, Deepa Nayar, and Srikumar Venugopal, Global Grids and Software Toolkits: A Study of Four Grid Middleware Technologies, High Performance Computing: Paradigm and Infrastructure, Laurence Yang and Minyi Guo (editors), Wiley Press, New Jersey, USA, June 2005.
4. JarekNabrzyski, Jennifer M. Schopf, Jan Weglarz, Grid Resource Management: State of the Art and Future Trends , (International Series in Operations Research & Management Science), Springer; First edition, 2003

DATA ANALYTICS USING PYTHON LAB

Objectives:

This Lab course will help students to work on python language and gain
1. Gain knowledge to work on Python Programming.
2. Gain Knowledge to solve Data science task through Python Programming.
3. Gain Knowledge to solve Big Data Analytics task through Python Programming.

LIST:-

1. Write a program to perform string operation using python programming.
2. Write a program to perform Function operation using python programming.
3. Write a program to perform List and Tuples using python programming.
4. Write a program to perform Dictionaries using python programming.
5. Write a program using Spicy and Numpy for Data Analytics using python programming.
6. Write a program using Pandas Data Frames for Data Analytics using python programming.
7. Write a program using Pandas Data Frames for Data Analytics using python programming.
8. Write a Program to classify the image using machine Learning by Scikit package using python programming.
9. Write a Program to classify the image using machine Learning(Deep Learning) by Scikit package using python programming.

OUT COMES:

After completed this lab the student will able.

- Develop program or software in the area of Data Science Field .
- Develop program or software in the area of Data Analytics Field .