# UNIVERSITY OF KERALA

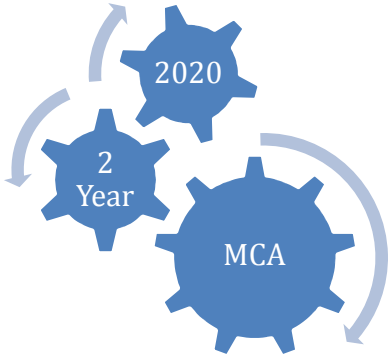
# Master of Computer Applications (MCA) [2 years]

## Regulation, Curriculum Framework and Syllabus

## (Outcome Based Education)

## (With effect from 2020 onwards)

## (A) REGULATION




I. **Introduction:** This regulation may be named as University of Kerala, Regulations for MCA, 2020 and is subject to the provisions of the Kerala University Act, 1974 and the statutes and ordinances if any issued in the subject is applicable from time to time. This regulation shall be applicable for students admitted from 2020 onwards.

II. **Objectives:** To create skilled man power at the level of Programmer, System Analyst, Project Manager and System Manager.

III. **Duration:** The duration of the course is four semesters in 2 years.

IV. **Eligibility:**

Candidates should have passed BCA/Bachelor Degree in Computer Science /Engineering or equivalent Degree recognized by the University of Kerala.

*OR*

Passed B.Sc./B.Com/BA with Mathematics at 10 + 2 level or at Graduation level or any equivalent degree recognized by the University of Kerala with additional bridge course as per the norms of the University of Kerala to apply for the programme and obtained at least 50% marks (45% marks in case of candidates belonging to reserved category) in the qualifying examination.

*Candidates seeking admission must ensure the eligibility of their qualifying degrees/programmes/courses by the University of Kerala prior to admission.*

**Bridge Courses**: Students who haven't undergone Computer Science as a core or complementary course for their undergraduate program has to study these courses in first semester itself and appear for the exam conducted by the University. A minimum pass in these courses are required for such candidate to register for the 3rd semester of the program and also for the publishing 1st & 2nd semester result. The student will get one chance in both the first and second semester of the program to attend the examination for the bridge courses. But, these marks will not be counted for finding class for the MCA programme. Institutions must ensure the conduct of the Bridge course suggested in the regular mode.

| Course Code | Bridge Courses | Credits |
|---|---|---|
| **MCA20B01** | Principles of Programming | 3 |
| **MCA20B02** | Digital Logic and Computer Architecture | 3 |
| **MCA20B03** | Programming C Lab | 2 |
| | **Total** | **8** |

**Question Paper Pattern (Bridge courses)**: The maximum mark for the theory examinations will be 50 and the time duration will be 2 hours. The minimum pass mark will be 40% for individual courses and 50% for consolidated marks of 3 courses. The question paper shall contain 3 parts; Part-A, Part-B and Part-C. Part-A shall be for 10 marks and contains 10 compulsory questions (MCQ/One-word/etc.), of 1 mark, 2 questions from each module. Part-B shall be for 20 marks and shall contain 15 short answer questions, 3 questions of 2 marks from each module out of which 2 questions from each module is mandatory. Part-C shall be for 20 marks and shall contain 10 questions, 2 questions of 4 marks from each module out of which 1 question from each module is mandatory. The lab examination will be for 50 marks, conducted internally by the institution monitored by the head of the Institution/Department, evaluating the skill of candidate.

| | One Module | | | | Complete Question set (5 modules) | | |
|---|---|---|---|---|---|---|---|
| Section | No. of Questions | No. of questions to attend (mandatory) | Marks for each question | Total Marks | No. of Questions | No. of questions to attend (mandatory) | Total Marks |
| Part-A | 2 | 2 | 1 | 2 | 10 | 10 | 10 |
| Part-B | 2 | 2 | 2 | 4 | 10 | 10 | 20 |
| Part-B | 2 | 1 | 4 | 4 | 10 | 5 | 20 |
| | | Marks distributed for a module : 10 | | | Total Marks : 50 | | |

V. **Pattern:** The student has to take, generally, five theory papers and two practical courses in the first three semesters. MOOC courses are included in first and fourth semester. Seminar and case study is included in second and third semesters respectively. In fourth semester the student has to undergo a major project work. In each week a student is supposed to get 11 practical hours and hence in every semester a total of 176 hours of practical training in the laboratories. The total contact hours for theory / tutorial /practical comes to around 30 hours/week. The attendance in the theory & practical is compulsory.

VI. **Examinations:** University Examinations will be conducted at the end of each semester as per the scheme included in this document.

Pass Requirements and provisions for classification of successful candidates.

a) A candidate shall be declared to have passed the semester examination in full if the candidate secures not less than 40% marks in written examination and not less than 50% marks in written (University) plus sessional marks put together in each paper. This rule applies to practical also. For the courses which have only sessional marks, a minimum of 50% is required for a pass; otherwise the student has to repeat that semester.

b) For a pass in main project the student has to obtain minimum 50% marks in internal evaluation and 50% marks in external evaluation. Otherwise the candidate has to repeat the $4^{th}$ semester.

c) If a student fails in one or more courses, he/she needs to reappear only in those courses. The rules for supplementary examinations will be same as that of the existing regulations.

d) Classification of (Pass) results into Distinction, I-Class, II-Class, etc. shall be as per the scheme prior to 2020 admissions.

**e)** Sessional Marks: The sessional marks are awarded based on 2 class tests and assignments/lab reports for theory / practical. Split up is shown below:

**Theory**

| Assignments (minimum2) | Class tests (minimum2) |
|---|---|
| 50% | **50%** |

**Practical:**

| Performance in the lab (Lab reports and experiments) | Lab tests (minimum2 |
|---|---|
| 50% | **50%** |

**Main Project**

| Topic | Performance | Evaluation |
|---|---|---|
| 10% | **40%** | **50%** |

For Seminars, the sessional marks are based on presentation/seminar report and participation. The students are required to present the progress of the main project work twice to the Department Faculty.

**VII.** Question Paper Pattern (Other than Bridge courses): The maximum mark for the theory examinations will be 100 and the time duration will be 3 hours. The question paper shall contain two parts; Part-A and Part-B. Part-A shall be for 40 marks and shall contain 10 compulsory short answer questions, 2 questions from each module with 4 marks. Part B shall be for 60 marks and shall contain 15 questions, 3 questions from each module out of which the student has to answer 2 questions from each module and has 6 marks for each question.

| | One Module | | | | Complete Question Set (5 modules) | | |
|---|---|---|---|---|---|---|---|
| Section | No. of Questions | No. of questions to attend (mandatory) | Marks for each question | Total Marks | No. of Questions | No. of questions to attend (mandatory) | Total Marks |
| Part-A | 2 | 2 | 4 | 8 | 10 | 10 | 40 |
| Part-B | 3 | 2 | 6 | 12 | 15 | 10 | 60 |
| **Marks distributed for a module : 20** | | | | | **Total Marks : 100** | | |

**VIII.** Each student has to successfully complete one MOOC course in the first semester and fourth semester from the topics related to (Communicative English, Research Methodology, Technical Writing, Entrepreneurship, Environmental studies, Cyber law or any new topic/technology introduced recently. etc.) with a minimum period of eight (08) weeks offered through E-learning platforms like SWAYAM, Coursera, etc.

## II. CURRICULUM FRAMEWORK

**Semester I**

| Course Code | Name of Course | Credits | Duration in Hours | | | Marks | | |
|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Sessional | Written/ Practical | Total |
| MCA20C11 | Mathematical Foundations for Computing | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C12 | Advanced Operating Systems | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C13 | Data Structures using Java | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C14 | Object Oriented Software Engineering | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C15 | Theory of Computation | 3 | 3 | 0 | - | 50 | 100 | 150 |
| MCA20P11 | Lab 1 –Data Structures using JAVA | 2 | - | - | 4 | 50 | 100 | 150 |
| MCA20P12 | Lab 2 – UML | 2 | - | - | 4 | 50 | 100 | 150 |
| MCA20G1X | MOOC-1 | 1 | - | - | 3 | 50 | - | 50 |
| | **TOTAL** | **20** | **15** | **4** | **11** | **400** | **700** | **1100** |

**Semester II**

| Course Code | Name of Course | Credits | Duration in Hours | | | Marks | | |
|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Sessional | Written/ Practical | Total |
| MCA20C21 | Python Programming | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C22 | Networking with TCP/IP | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C23 | Database Management Systems | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C24 | Design & Analysis of Algorithms | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20E2X | Elective -1 | 3 | 3 | - | - | 50 | 100 | 150 |
| MCA20P21 | Lab1 – Python | 2 | - | - | 4 | 50 | 100 | 150 |
| MCA20P22 | Lab2 – DBMS | 2 | - | - | 4 | 50 | 100 | 150 |
| MCA20G21 | Seminar | 1 | - | - | 3 | 50 | - | 50 |
| | **TOTAL** | **20** | **15** | **4** | **11** | **400** | **700** | **1100** |

**Semester III**

| Course Code | Name of Course | Credits | Duration in Hours | | | Marks | | |
|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Sessional | Written/ Practical | Total |
| MCA20C31 | Machine Intelligence | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C32 | Mobile Application Development | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20C33 | Data Science with Big Data Analytics | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20E3X | Elective -2 | 3 | 3 | 1 | - | 50 | 100 | 150 |
| MCA20E3X | Elective -3 | 3 | 3 | - | - | 50 | 100 | 150 |
| MCA20P31 | Lab 1 - Mobile Application Development | 2 | - | - | 4 | 50 | 100 | 150 |
| MCA20P32 | Lab 2 –ML | 2 | - | - | 4 | 50 | 100 | 150 |
| MCA20G31 | Case study (SE/ML/DS/Mobile App) | 1 | - | - | 3 | 50 | - | 50 |
| | **TOTAL** | **20** | **15** | **4** | **11** | **400** | **700** | **1100** |

**Semester IV**

| Course Code | Name of Course | Credits | Duration in Hours | | | Marks | | |
|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Sessional | Written/ Practical | Total |
| MCA20C41 | Project Work | 12 | - | - | 27 | 50 | 150 | 200 |
| MCA20C42 | Comprehensive Course Viva | 2 | - | - | - | - | 50 | 50 |
| MCA20G4X | MOOC-2 | 1 | - | - | 3 | 50 | - | 50 |
| | **TOTAL** | **15** | | | **30** | **100** | **200** | **300** |

| Elective Courses | | Credits |
|---|---|---|
| **Elective I** | | |
| MCA20E21 | Probability and Statistics | 3 |
| MCA20E22 | Block Chain Technology | 3 |
| MCA20E23 | Principles of Management | 3 |
| MCA20E24 | Cloud Computing | 3 |
| **Elective II** | | |
| MCA20E31 | Advanced Machine Learning Models | 3 |
| MCA20E32 | Biologically Inspired Algorithms | 3 |
| MCA20E33 | Bioinformatics | 3 |
| MCA20E34 | Cyber Forensics | 3 |
| **Elective III** | | |
| MCA20E35 | Social Network Analysis | 3 |
| MCA20E36 | Digital Marketing | 3 |
| MCA20E37 | Internet of Things | 3 |
| MCA20E38 | Cyber Security & Cryptography | 3 |

## PROGRAMME OUTCOMES (PO) for MCA Programme

| | PROGRAMME OUTCOMES (PO) |
|---|---|
| PO1 | Demonstrate the computational intelligence, computing and management ability to apply these to professional careers |
| PO2 | Recognize the necessity, and have the ability, to contribute for the development of the society |
| PO3 | Promote continuous life-long learning, team work and through education should be able to appreciate human values and ethics. |
| PO4 | Able to articulate the knowledge of professional ethics and cyber regulations, responsibilities and norms of professional computing |
| PO5 | Able to use the logical thinking and critical analysis for problem solving and evaluation |

## PROGRAMME SPECIFIC OUTCOMES (PSO) – for the MCA programme

| | PROGRAMME SPECIFIC OUTCOMES (PSO) |
|---|---|
| PSO1 | Demonstrate technical skill and domain knowledge to develop solutions for addressing the real world issues |
| PSO2 | Applying entrepreneurial skills and software management skills augmented with a rich set of communication, teamwork and leadership skills to excel in their profession |
| PSO3 | Upgrade the skill set to current industry standards to make the students industry ready professionals. |
| PO4 | Obtain Practical hands on experience in programming languages including python, Java to solve problems |
| PO5 | Provide opportunity for the students to develop the entrepreneurial skills and innovative contributions with their start-ups. |
| PO6 | Apply scientific knowledge of computing fundamentals and domain knowledge appropriate for resolving the real world problems. |

| Semester I | Course Code: MCA20C11 | Credits: 3 |
|---|---|---|

## MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Compute a generating function and apply them to combinatorial problems |
| CO2 | Apply the inclusion/exclusion principle |
| CO3 | Solve linear algebra problems with linear equations, matrix calculus and vectors |
| CO4 | Discuss the usage of geometric transformations. |
| CO5 | Illustrate different decomposition methods used in linear system of equations |
| CO6 | Explain unconstrained optimization and Linear Programming Problems |

### COURSE CONTENT

**MODULE I:** Elementary Combinatorics: Basis of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, the principles of Inclusion - Exclusion. Pigeon hole principles and its application.

**MODULE II:** Linear algebra: Matrices, vectors and determinants, Eigen values, Eigen vectors, Eigen value problems, vector differential calculus-Inner product, cross product, gradient of a scalar field, divergence of a vector field and curl of a vector field.

**MODULE III:** Geometric transformations-Translations, Rotation around the origin, Rigid motions and homogeneous representations, Affine transformations, Coordinate Transformation on Image Arrays.

**MODULE VI:** Numeric Analysis: Introduction, solution of equations by iteration, numeric linear algebra-Linear Systems: Gauss Elimination, LU factorization, matrix inversion, Least squares method.

**MODULE V:** Optimization: Basic concepts, Unconstrained Optimization-method of Steepest Descent, Linear Programming-Normal, pivotal reduction of a general system of equations, simplex method.

### REFERENCES


- Discrete mathematics for computer scientists & mathematicians JL Mott, A Kandel, TP Baker PHI.
- Discrete Mathematical structures Theory and application-Malik &Sen, Course Technology, 2004
- Ernest Davis,Linear Algebra and Probability for Computer Science Applications,CRC Press,978-1-4665-0159-1
- Erwin Kreyszig, Advanced Engineering Mathematics (10th Edition), 2011 John Wiley & Sons , ISBN-13: 978-0-571-72897-9
- Michael Baron,Probability and statistics For computer scientists(2nd edition), Chapman and Hall/CRC, ISBN 978-0-570-55836-5

| Semester I | Course Code: MCA20C12 | Credits: 3 |
|---|---|---|

## ADVANCED OPERATING SYSTEMS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Explore the fundamental concepts of different models of Operating systems |
| CO2 | Understand the design and functions of operating systems. |
| CO3 | Illustrate different process scheduling models in Operating Systems |
| CO4 | Differentiate different types of OS and its significance. |
| CO5 | Understand the file and memory management in UNIX Operating Systems. |
| CO6 | Understand the need and importance of Parallel systems. |
| CO7 | Understand the working of Distributed Systems. |
| CO8 | Compare Real Time OS, Mobile OS and Multi-processor OS. |

### COURSE CONTENT

**MODULE I:** Functions of operating systems, Design approaches: layered, kernel based and virtual machine approach, types of advanced operating systems (NOS, DOS, Multiprocessor OS, Mobile OS, RTOS, Cloud OS)

**MODULE II:** *File Management*- System Structure, User Perspective, Architecture - Buffer cache - File Representation: Structure of file Directories. *Memory Management*- Detailed design of Process Structure: Kernel Data structures for process. Context of a Process: Static and Dynamic. *Parallel Systems and computing*- Shared memory machines, Synchronization, Communication, Shared memory multiprocessor OS.

**MODULE III:** *Distributed Operating system concepts*- Goals, Distributed Computing Models, Hardware and Software Concepts, Architecture of DOS. Design Issues, Distributed communication, shared memory, synchronization - Distributed Object based System.

**MODULE IV:** *Multiprocessor Operating System*- **I**ntroduction, Basic multiprocessor system architectures, design issues, Threads, Process synchronization: the test and set instruction, the swap instruction, implementation of the process wait. Processor scheduling: Issues, Co-scheduling, Smart scheduling, Affinity Based scheduling

**MODULE V:** *Real Time OS*- Characteristics of Real Time operating Systems, Classification of Real Time Operating Systems, Scheduling in RTOS: Clock driven: cyclic, Event driven: EDF and rate monotonic scheduling. *Mobile OS*- Architecture, Android OS, iOS, Virtual OS, Cloud OS and their design issues.

## REFERENCES


- Bhatt P.C.P., An Introduction to Operating Systems: Concepts and Practice, 3/e, Prentice Hall of India, 2010.
- William Stallings, Operating Systems: Internals and Design Principles, Pearson, Global Edition, 2015.
- Mukesh Singhal and Niranjan Shivarathri, Advanced Concepts in Operating Systems, McGraw-Hill Series in Computer Science,1993.
- Andrew S. Tanenbaum, Distributed Operating Systems, Addison – Wesley, 1998
- Jean Bacon, Concurrent Systems, Addison – Wesley, 1998
- William Stallings, Operating Systems, Prentice Hall, 1995.
- Pradeep K Sinha, “Distributed Operating Systems: Concepts and design”, PHI, 2007.

| Semester I | Course Code: MCA20C13 | Credits: 3 |
|---|---|---|

## DATA STRUCTURES USING JAVA

### COURSE OUTCOMES

| | |
|---|---|
| **CO1** | List the advantages of the object oriented programming approach |
| **CO2** | Explore the basic programming concept of Java |
| **CO3** | Understand the advanced programming capabilities of Java |
| **CO4** | Classify the types of data Structures |
| **CO5** | Explain the algorithms associated with each type of data structures |
| **CO6** | Identify the appropriate data structures to solve different problems |
| **CO7** | Implement the algorithms using the advanced features of Java |

### COURSE CONTENT

**MODULE I:** Basic Concepts of Java: Object Oriented Programming, Features of Java, Classes, Interfaces, Constructors and Finalizers, Packages, Exception Handling, API.

**MODULE II:** Advanced Concepts in Java: Multithreading, Event Handling, Applets, Graphics, Text, AWT Controls, Layout Manager and Menus. I/O Streams, File Class and Operations.

**MODULE III:** Data Structures: Introduction, Abstract Data Types, Linear and Non Linear Data Structures, Searching: Linear Search, Binary Search, Implementation in Java, Sorting: Insertion Sort, Bubble sort, Selection Sort, Merge Sort, Quick Sort, Implementation in Java, Stacks: Representation, Implementation of Stack Operations in Java, Queues: Representation, Implementation of Queue Operations in Java.

**MODULE IV:** Linked Lists: Representations, Linear Linked List, Doubly Linked List, Circular Linked List, Implementation of operations on Linked Lists in Java, Hashing : Hash functions, Hash Tables, Chaining.

**MODULE V:** Trees: Representation, Operations on Trees, BST, AVL Trees, Red-Black Trees, B-Trees, Implementation of Trees in Java, Graphs: Representation, Operations on Graphs, Spanning Trees, implementation of Graphs in Java.

## REFERENCES

### Text books


1. John Hubbard, Data Structures with Java, 2ed (Schaum's Outlines)
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein,“Introduction to Algorithms”, MIT Press
3. Paul J. Deitel and Harvey Deitel , Java How to Program
4. Schildt Herbert, Java: The Complete Reference
5. Michael T. Goodrich, Data Structures and Algorithms in Java


### Additional and Web –Resources


1. https://onlinecourses.nptel.ac.in/noc20_cs85/preview
2. https://enos.itcollege.ee/~jpoial/algorithms/GT/Data%20Structures%20and%20Algorithms%20in%20Java%20Fourth%20Edition.pdf

| Semester I | Course Code: MCA20C14 | Credits: 3 |
|---|---|---|

## OBJECT ORIENTED SOFTWARE ENGINEERING

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Discuss about the steps in Unified Approach. |
| CO2 | Draw UML diagrams including class diagram, activity diagram, use case diagram and sequence diagram for a given problem statement |
| CO3 | Illustrate the steps in Object Oriented Analysis and Object Oriented Design. |
| CO4 | Explain the principles and practices in Agile Software development methodology. |
| CO5 | Compare software quality assurance techniques. |
| CO6 | State Myer's principle for debugging. |
| CO7 | Design an object oriented system to solve any real life problem. |

### COURSE CONTENT

**MODULE I:** Introduction – Object Oriented Systems development life cycle. Object oriented Methodologies- Booch's Methodology - Rambaugh's Methodology, Jacobson's Methodology- Patterns and Frameworks.

**MODULE II:** Fundamentals of Object Oriented design using Unified Modeling Language, UML- Use case diagram- Class diagram- Sequence diagram- collaboration diagram-State chart diagram- Activity diagram- Component diagram- deployment diagram.

**MODULE III:** Object oriented analysis: Use Case Model- Identifying use cases identifying actors- Documentation- Object analysis - Classification-different approaches- Identifying Classes- Identifying Object relationships – Attributes and Methods.

**MODULE IV:** Object oriented design: Design process- Design axioms - Corollaries- Design Patterns- Designing Classes - Designing Protocols and Class visibility - Defining Attributes- Designing Methods-Guidelines for identifying bad design.

**MODULE V:** Agile Software Development - Agile Practices & Principles- Software Quality Assurance- Bugs and Debugging- Testing Strategies- Developing test cases- Developing test plans- Debugging Principles- Introduction to Agent Oriented Software Engineering.

## LEARNING RESOURCES

## REFERENCES

### TEXT BOOKS


- Ali Bahrami, Object Oriented Systems Development, Tata McGraw-Hill, 1999
- Martin Fowler, UML Distilled, Second Edition, PHI/Pearson Education, 2002.
- James Rumbaugh, Ivar Jacobson, Grady Booch, The Unified Modelling Language Reference Manual, Addison Wesley, 1999.
- Stephen R. Schach, Introduction to Object Oriented Analysis and Design, Tata McGraw-Hill, 2003.
- Hans-Erik Eriksson, Magnus Penker, Brain Lyons, David Fado, UML Toolkit, OMG Press Wiley Publishing Inc., 2004.
- A survey of Agent-Oriented Software Engineering. Amund Tveit. Norwegian University of Science and Technology. May 8, 2001.
- Teach Yourself UML in 24 Hours, Joseph Schmuller, 3rd Edition, ISBN 81-297-0609-1, Pearson Education, 2004
- Grady Booch, James Rumbaugh, Ivar Jacobson, “UML User Guide”, Addison Wesley, 2002.
- Craig Larman (2005) Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition,
- Lethbridge and Laganiere, "Object-Oriented Software Engineering: Practical software development using UML and Java"

| Semester I | Course Code: MCA20C15 | Credits: 3 |
|---|---|---|

## THEORY OF COMPUTATION

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Exploring the basics of Language |
| CO2 | Construction of Deterministic & Non Deterministic Automata |
| CO3 | Exploring Grammars – Ambiguity - Regular Expressions & Regular Grammars |
| CO4 | Discussing Context Free Languages & Greibach forms |
| CO5 | Discussing Pumping Lemma & construction of Push down automata |
| CO6 | Exploring various types of Turing Machines – Mealy & Moore machines |
| CO7 | Discuss Decidability & Halting Problem |

### COURSE CONTENT

**MODULE I :** Concepts of Automata Theory : Strings, Alphabet, Language- Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, Finite Automata: NFA , DFA , Finite Automata with Epsilon transitions NFA to DFA conversion, minimisation of DFA, Finite Automata with Epsilon Transition- Finite Automata with output- Moore and Mealy machines.

**MODULE II:** Regular Expressions & Languages: Regular expressions, Finite Automata & Regular Expression operations, Conversion of Finite Automata to Regular expressions. Converting Regular Expressions to Automata - Algebraic Laws for Regular Expressions. Pumping Lemma for Regular Languages, Application of Regular Expressions

**MODULE III:** Context free grammar: Derivation trees, and sentential forms. Parse Trees - Ambiguity in Grammars, Chomsky normal form, Greibach normal form, Pumping Lemma for Context Free Languages. Enumeration of properties of Context Free Language, PDA ,Acceptance of Context Free Language, Deterministic PDA ,Application of CFG

**MODULE IV:** Turing Machines: Definition , Transition diagram, Design & Roles of Turing machine, Church-Turing Thesis, Modular Construction of complex Turing machines, Types of Turing machines Extensions of Turing machines, Non-Deterministic Turing Machines. Restricted Turing Machines

**MODULE V**

Complexity Theory: Intractable Problems: Definition of P and NP problems, NP complete and NP hard problems. Decidability and Undecidability - Decidability & Halting problems - Undecidability and Reducibility in TOC,

## REFERENCES

### Text Books


1. John E Hopcroft ,Rajeev Motwani and Jeffrey D Ullman " Introduction to Automata Theory, Languages, and Computation ", Third Edition Pearson
2. Michael Sipser " Introduction to the Theory of Computation " Cengage Publishers ,2003
3. John C Martin , Introduction to Languages and he Theory of Computation , TMH 2013


### Reference Text :


1. Dexter C Kozen , Automata and Computatbility , Springer 1999

| Semester I | Course Code: MCA20P11 | Credits: 2 |
|---|---|---|

## LAB 1 – DATA STRUCTURES USING JAVA

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Acquire knowledge in the object oriented programming concept |
| CO2 | Design the algorithms to match with the given problem specifications |
| CO3 | Implement linear and non linear data structures using the advanced features of Java |
| CO4 | Familiarize the debugging concept in Java Code |
| CO5 | Formulate real world applications with the help of appropriate algorithms |
| CO6 | Generate different test cases for testing the validity of the developed programs |
| CO7 | Write technical report based on the results of the experiments |

### COURSE CONTENT

**List of Experiments:**

1. Searching : Implement Linear and binary search
2. Sorting: Implement Insertion Sort, Bubble sort, Selection Sort, Merge Sort, Quick Sort
3. Stacks: Perform stack operations
4. Applications of Stacks: Perform evaluation of expressions
5. Queues: Perform queue operations.
6. Application of queues: Simulate a queue in a real life situation
7. Linked List : Implementation of linked lists
8. Hashing : Apply hashing functions for searching
9. Trees: Familiarize the different operations on Trees
10. BST : Implement a BST
11. AVL Trees: Implement an AVL tree
12. Red-Black Trees: Implement Red-Black Trees
13. B-Trees: Generate B-Trees and perform the operations
14. Graphs – Perform the graph traversals
15. Spanning Trees : Implementation of Spanning trees

## REFERENCES

### Text books


1. Lab manual for Data structures through Java, V V Muniswami
2. Java Lab Manual by Madhu Mathi
3. Data Structures in Java  A laboratory Course, Sandra Andersen


### Additional and Web -Resources


1. https://www.udemy.com/course/data-structures-and-algorithms-in-java/
2. https://onlinecourses.nptel.ac.in/noc20_cs85/preview
3. https://enos.itcollege.ee/~jpoial/algorithms/GT/Data%20Structures%20and%20Algorithms%20in%20Java%20Fourth%20Edition.pdf

| Semester I | Course Code: MCA20P12 | Credits: 2 |
|---|---|---|

## LAB 2 – UNIFIED MODELLING LANGUAGE (UML)

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Implement basic SQL commands including creation of database, tables and insertion, updation, deletion and selection of table contents. |
| CO2 | Perform joining of tables and implementing nested queries. |
| CO3 | Draw UML diagrams including class diagram, use case diagram, activity diagram, collaboration diagram, sequence diagram and state chart diagram. |

### COURSECONTENT

Lab exercises related with the following should be implemented in this course.

1. Implementing UML diagrams
    a. Class diagram
    b. Use case diagram
    c. Activity diagram
    d. State chart diagram
    e. Sequence diagram
    f. Collaboration diagram

**Sample Problem Statement**

There is a need to develop software for tracking the library loan records. Library patrons may borrow books, magazines, compact discs and audio tapes. A library must manage each copy of a library item. For eg., a library may have five copies of the book “Wings of the Fire”. Each borrower has library card. Each type of library item has a check out period and a maximum number of renewals. For example the children’s books may be checked out for a month while books for the adults may be checked only for two weeks. Ordinary books can be renewed once; books with a pending request may not be renewed. The system must record the actual return date and any fine that was paid. Draw the UML diagrams for the given problem statement.

a) Draw Class Diagram
b) Draw Use Case diagram
c) Draw Activity diagram
d) Draw Sequence Diagram

| Semester I | Course Code: MCA20G1X | Credits: 1 |
|---|---|---|

## MOOC 1

Each student has to successfully complete one MOOC course in the first semester from the topics related to (Communicative English, Research Methodology, Technical Writing, Entrepreneurship, Environmental studies, Cyber law or any new topic/technology introduced recently. etc.) with a minimum period of eight (08) weeks offered through E-learning platforms like SWAYAM, Coursera, etc..

| Semester II | Course Code: MCA20C21 | Credits: 3 |
|---|---|---|

## PYTHON PROGRAMMING

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Explore the programming skills of students by imparting both procedural and object oriented concepts |
| CO2 | Analyse how to find solutions to real world problems using Python |
| CO3 | Implement exception handling and file management in python |
| CO4 | Perform complete software development including GUI programming |
| CO5 | Explore the packages and modules in Python |

### COURSE CONTENT

**MODULE I :**.Introduction to Python: Features of Python - Identifiers - Reserved Keywords, comments in python, Python Data Types, mutable and immutable objects, Data Type Conversion, operators, Expressions, Execution Control Structures, Python Standard Library, User-Defined Functions, Parameter Passing, Input , Output and Import Functions, Recursion, Function composition, Memory Management During Recursive Function Calls. Global versus Local Namespaces.

**MODULE II:** Containers: Strings and Lists, String Methods, Formatted Output, Two Dimensional Lists, Iterating through Two Dimensional Lists, Dictionaries, Tuples and Sets, Properties Operators and Methods of Containers. Lists as arrays. Other Built-in Container Types, Character Encoding and Strings, Errors and Exception Handling-Exception Types, Exception Handling using Try & Except. User Defined Exceptions.

**MODULE III:** Objects and Classes, Defining a Class in Python, Constructors. Classes as Namespaces Inheritance: Multiple and Multilevel Inheritance, Modifying Built in Classes Using Inheritance, Operator Overloading, Built-in Modules - Creating Modules - Import statement - Locating modules - Namespaces and Scope - The dir() function - The reload function - Packages in Python.

**MODULE IV:** GUI Programming- Introduction – Tkinter Widgets – Label – Message Widget – Entry Widget – Text Widget – tk Message Box – Button Widget – Radio Button- Check Button – Listbox Frames _ Toplevel Widgets – Menu Widget-event based Tkinter widgets.

**MODULE V:** NumPy: Creating Arrays (array() and arange), reshape(), sum(), min() and max() methods, Item wise arithmetic operations. Pattern Matching Using Regular Expressions: Python Standard Library Module, Database Programming in Python- Creating Tables, Querying

(Inserting Tuples, Selecting Rows and Updating Tuples) Using Cursor to Iterate over Selected Tuples, Files: Opening and Closing a File, Opening Modes, Various Read and Write Methods.

## REFERENCES

### Text books


- "Taming PYTHON By Programming", Jeeva Jose Khanna Publications 4.2
- Kenneth A. Lambert, "Fundamentals of Python: First Programs", CENGAGE Learning,2012.
- Paul Gries, Jennifer Campbell and Jason Montojo, "Practical Programming: An Introductionto Computer Science using Python 3", Second edition, Pragmatic Programmers,LLC,2013.
- Timothy A. Budd, "Exploring Python", Mc-Graw Hill Education (India) Private Ltd Beazley, D. M. (2009). Python essential reference. Addison-Wesley Professional.


### References


- Barry, P. (2010). Head First Python. " O'Reilly Media, Inc.".
- Punch, W. F., &Enbody, R. (2010). The practice of computing using python. Addison-Wesley Publishing Company.
- Mark, S. (2009). Programming in Python 3. Pearson Education India.
- jubomirPerkovic, "Introduction to Computing Using Python: An Application Development Focus", Wiley, 2012.
- Charles Dierbach, "Introduction to Computer Science Using Python: A Computational Problem-Solving Focus", Wiley, 2013.
- Allen B Downey, "Think Python" ,Oreilly, 2012
- Dr.Varghese Paul, Dr.AnjanaS.Chandran,"Introduction To Computing And Problem Solving Using Python",Educational Publishers And Distributors, 2016

| Semester II | Course Code: MCA20C22 | Credits: 3 |
|---|---|---|

## NETWORKING WITH TCP/IP

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Illustrate the phases in networking |
| CO2 | Compare various layered approaches in networking |
| CO3 | Analyse the efficiency of a network |
| CO4 | Define the boundaries between the layers in a network |
| CO5 | Investigate the reliability of TCP in data transmission. |

### COURSE CONTENT

**MODULE I:** Introduction, Uses of computer networks, Network hardware, Network Software. Reference models: The OSI Reference Model, The TCP/IP Reference Model. Example networks, The Internet, Third ,Fourth and Fifth Generation Mobile Phone Networks, Wireless LANs ,RFID and Sensor Networks.

**MODULE II:** Internetworking Concepts: Application-Level Interconnection, Network-Level Interconnection, Properties of the Internet, Internet Architecture, Interconnection through IP routers .Internet addressing: Universal Host Identifiers, The Original IPv4 Classful Addressing Scheme, Dotted Decimal Notation Used With IPv4,IPv4 Subnet Mask Representation And Slash Notation, The IPv6 Addressing Scheme, IP Addresses, Hosts, And Network Connections. ARP.

**MODULE III:** Internet Protocol: Purpose And Importance, Datagram Encapsulation, Time To Live (IPv4) And Hop Limit (IPv6),Forwarding In An Internet, Transmission across a single Network, The IP Forwarding Algorithm .The Internet Control Message Protocol, Error Reporting Vs. Error Correction, Testing destination reachability and status (Ping),Checksum computation.

**MODULE IV:** User Datagram Protocol (UDP),UDP Encapsulation and protocol layering .Reliable Stream Transport Service (TCP):Reliability: Acknowledgements and retransmission, The Sliding Window Paradigm, The Transmission Control Protocol, TCP Segment Format, Establishing and closing a TCP Connection, TCP State Machine, Silly Window Syndrome And Small Packets.

**MODULE V:** Routing: Automatic Route Propagation, Distance-Vector (Bellman-Ford) Routing, Routing Protocols. Network Virtualization: VPNs, NATs, and Overlays. Domain Name System (DNS),Mapping Domain Names To Addresses, Electronic Mail (SMTP, POP, IMAP, MIME),Internet Security And Firewall Design (IPsec, SSL),

## REFERENCES

**Text books:**


1. Andrew S. Tanenbaum,David J. Wetherall “Computer Networks”,Pearson
2. Douglas E Comer ,“Internetworking with TCP/IP”,Pearson


**References**


1. Larry L. Perterson and Bruce S. Davie , “Computer Networks- A Systems Approach”, Morgan Kaufmann
2. JochenSchiiler, “Mobile Communications”, Pearson,
3. Behrouz A. Forouzan,” TCP/IP Protocol Suite”,McGraw Hill

| Semester II | Course Code: MCA20C23 | Credits: 3 |
|---|---|---|

## DATABASE MANAGEMENT SYSTEMS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Explore the fundamental concepts of databases |
| CO2 | Construct an Entity-Relationship (E-R) model from requirement specifications. |
| CO3 | Perform the transformation of the conceptual model into corresponding logical data structures. |
| CO4 | Model and design a relational database following the design principles of normalisation. |
| CO5 | Develop queries for relational database in the context of practical applications |
| CO6 | Explain and illustrate fundamental principles of data organization and concurrent transaction processing. |
| CO7 | Explain the latest trends and technologies in database |
| CO8 | Estimate the storage size of the database and design appropriate storage techniques |
| CO9 | Explain the basic requirements for Backup and recovery |

### COURSE CONTENT

**MODULE I :**Introduction to Database Management System-The file system, Limitations of file system, The Database Approach, Schema, Instance, The Logical DBMS Architecture, Data Abstraction, Three level architecture of DBMS , Mappings between levels, Data independence, Physical DBMS Architecture, DML Pre-compiler, DDL Compiler, File Manager, Database Manager, Query Processor, Data files indices and Data Dictionary, Database Administrator, Database Users, Data Models

**MODULE II:** The Relational Model-Domains, Attributes, Tuple and Relation, Super keys Candidate keys and Primary keys for the Relations, Relational Constraints, Domain Constraint, Key Constraint, Integrity Constraint, Update Operations and Constraint Violations. Relational Algebra-Basic Set Operation, Cartesian product, Relational Operations, extended operations. Entity Relationship (ER) Model, Entities, Attributes, Relationships, Mapping constraints, Conversion of E-R Diagram to Relational Database.

**MODULE III:** Relational Database Integrity-The Keys, Referential Integrity, Entity Integrity; Functional Dependency, Transitive dependency, partial dependency, Multi-valued dependency, Normalization–1NF, 2NF, 3NF,4NF, 5NF, Boyce Codd Normal Form, Inference axioms, Desirable Properties of Decomposition, Lossy and Lossless decomposition, Attribute Preservation, Dependency Preservation, Lack of redundancy, Rules of Data Normalization, Eliminate Repeating Groups, Eliminate Redundant Data, Eliminate Columns Not Dependent on Key.

**MODULE IV:** The Structured Query Language–SQL; DDL, DML, DCL, Database Objects: Views, Sequences, Indexes and Synonyms, Table Handling Assertion and views, Cursors, triggers and stored procedures, Functions & Packages, Embedded SQL, dynamic SQL, Storage and File Structure, Indexing & Hashing, Transactions and Database Recovery -Transactions, Properties of a transaction, Concurrent Transactions, The Locking Protocol, Serializable Schedules, Locks, Two Phase Locking (2PL), Deadlock and its Prevention, Optimistic Concurrency Control, Recovery-Kinds of failures, Failure controlling methods, Database errors, Recovery Techniques, Security & Integrity.

**MODULE V:** Enhanced Database models-Object Oriented Database-Limitations of Relational databases, The need of Object oriented databases, Complex Data Types, Structured Types and Inheritance in SQL, Object-Oriented versus Relational databases; Database and XML-Structured Semi structure and unstructured data, XML hierarchical tree data model, Documents DTD and XML schema, XML Documents & Database, XML querying and transformation, Storage of XML data, XML database applications.

## REFERENCES

### Text books


- Ramon A. Mata-toledo and Pauline K. Cushman, Database Management Systems Schaum's Outlines, Tata McGraw Hill
- Henry F. Korth, Sudarshan and Abraham Silbershatz- Database System Concepts-6thEdn, McGraw Hill, 2010.
- C.J.Date, Longman, Dr.S.Swamynathan, Introduction to Database Systems, Pearson Education – 2010
- Thomas Connolly and Carolyn Begg - Database systems, 4th edition – Pearson Education, 2009
- R. Narang – Database Management System, PHI4. Hansen and Hansen – Database Management and Design, 2ndedition, PHI
- Atul Kahate, Introduction to Data Base Management Systems, Pearson Education
- Ramez Elmasri, Shamkant B. Navathe - Fundamentals of Database Systems –Fifth Edition - Addison Wesley Higher Education – 2010
- Raghu Ramakrishnan, Johannes Gehrke - Database Management Systems - Third Edition - McGraw-Hill – 2014
- Hoffer, Prescott & McFadden - Modern Database Management – Eighth Edition - Prentice Hall – 2010
- Kifer, Bernstein & Lewis - Database Systems: An Application Oriented Approach, Compete Version – Second Edition - Addison Wesley Higher Education – 2010

| Semester II | Course Code: MCA20C24 | Credits: 3 |
|---|---|---|

## DESIGN AND ANALYSIS OF ALGORITHMS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Analyze the asymptotic performance of algorithms |
| CO2 | Solve recurrences problems using Recursion tree method -Master method |
| CO3 | Demonstrate a familiarity with major algorithms and data structures |
| CO4 | Apply important algorithmic design paradigms and methods of analysis |
| CO5 | Verify the proof of rigorous correctness for algorithms |
| CO6 | Synthesize efficient algorithms in common engineering design situations. |
| CO7 | Classify computational problems to P, NP, NP Hard and NP Complete |

**MODULE-I :** The role of computing , Insertion sort, Analyzing algorithms, Designing algorithms, Growth of functions-Asymptotic notations, Standard notations , Divide and Conquer method , maximum subarray problem, Strassen's algorithm for matrix multiplication, Substitution method, Recursion tree method -Master method for solving recurrences, Proof of the master theorem, Probabilistic analysis and randomized algorithms, The Hiring problem, Indicator random variables, Randomized algorithms, Probabilistic analysis

**MODULE-II :** Heap sort, Maintaining the heap property, Building a heap, Algorithm, Priority queues, Quick sort, Description, Performance, Randomized version of quick sort, Analysis of quick sort, Sorting in Linear time, Lower bounds for sorting, Counting Sort, Radix sort, Bucket sort, Medians and order statistics, Minimum and Maximum, Selection In expected linear time&in worst-case linear time

**MODULE-III :** Hash tables, Direct address tables, Hash functions, Open addressing, Perfect hashing, Binary search trees, Randomly built binary search trees, Red-Black tree-Properties, Insertion, Deletion, Rotations, Augmenting Data Structures-Dynamic order statistics, Interval trees, Dynamic programming- Rod cutting, Matrix chain multiplication, Longest common subsequence, Optimal binary search trees

**MODULE-IV :** B-Trees, Fibonacci Heaps, van Emde Boas trees, Data structures for disjoint sets, Graph algorithms-BFS, DFS, Topological sort, strongly connected components, Growing a minimum spanning tree, Kruskal's and Prim's Algorithm, Bellman-Ford algorithms, Single source shortest paths in DAGs, Dijkstra's algorithm, Proof of shortest path properties, The Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs, Flow networks – The Ford Fulkerson method, Maximum bipartite matching, Push-relabel algorithms

**MODULE-V :** Basics of dynamic multithreading, multithreaded matrix multiplication, multithreaded merge sort, symmetric positive-definite matrices and least squares approximation, The simplex algorithm, Duality, String-Matching - Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm. Greedy algorithms- Elements of the greedy strategy. Complexity Theory -. P & NP classes -NP-Hard & NP-Complete Classes

## REFERENCES

**Textbooks:**


1. Cormen, Thomas H, Leiserson, Charles E & Rivest, Ronald L, 'Introduction to Algorithms ', Prentice Hallof India Private Limited, New Delhi, Third Edition, 2009
2. Horowitz, Sahni &Rajasekharan, ' Computer Algorithms- Silicon press, 2nd edition, 2008
3. Aho , Hopcroft, Ullman ,'Design &Analysis of Computer Algorithms'


**References:**


1. Robert Sedgewick, Kevin Wayne , 'Algorithms ' , Pearson Education, 2011
2. Sahni, ' Data Structures, Algorithms and Applications in C++ ' , Silicon Press, 2nd edition, 2004
3. John Kleinberg and Eva Tardos, ' Algorithm Design ' AW[2005]
4. Anany C. Levitin. ' Introduction to the Design & Analysis of Algorithms ' Second edition
5. Dasgupta, Papadimitrou and Vazirani, ' Algorithms ' McGraw-Hill Education, 2006.

| Semester II | Course Code: MCA20P21 | Credits: 2 |
|---|---|---|

## LAB – 1 PYTHON

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Implement simple Python Programs with conditions and loops |
| CO2 | Implement functions in Python |
| CO3 | Represent compound data using Python lists, tuples and dictionaries. |
| CO4 | Read and write data from/to files in Python. |
| CO5 | Implement Programs for creating and handling of modules and packages |
| CO6 | Implement GUI programming |
| CO7 | Implement Programs involving a variety of Exception Handling situations |

### COURSE CONTENT

1. Implement Simple python programs using decision making and looping control structures
2. Implement Programs using functions
3. Recursive functions
4. Programs using one dimensional arrays(mean, median, mode)
5. Programs using two dimensional arrays
6. Linear search
7. Binary Search
8. Operations using string, tuple, list, dictionary and set data types(any 10 each)
9. Implement Bubble sort, insertion sort, selection sort
10. Demonstrate polymorphism-operator and function overloading
11. Demonstrate exception handling
12. Implement inheritance
13. Demonstrate creating modules
14. Packages in python
15. Implement GUI Programming using all widgets
16. Usage of NumPy
17. Implement database programming by creating tables and querying tuples
18. Implementation of Files in Python

## REFERENCES

**Text Books:**


- “Taming PYTHON By Programming”, Jeeva Jose Khanna Publications 4.2
- Kenneth A. Lambert, “Fundamentals of Python: First Programs”, CENGAGE Learning,2012.
- Paul Gries, Jennifer Campbell and Jason Montojo, “Practical Programming: An Introductionto Computer Science using Python 3”, Second edition, Pragmatic Programmers,LLC,2013.
- Timothy A. Budd, “Exploring Python”, Mc-Graw Hill Education (India) Private Ltd

| Semester II | Course Code: MCA20P22 | Credits: 2 |
|---|---|---|

## LAB – 2 : DBMS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Design a database for a given problem using database design principles. |
| CO2 | Implement stored programming concepts (PL-SQL) using Cursors and Triggers. |
| CO3 | Identify the relationship between tables |
| CO4 | Implement privileges to different categories of database users |
| CO5 | Create procedures, functions, packages, views and assertions |
| CO6 | Apply transaction control using commands |

### COURSE CONTENT

1. Creation of a database using DDL commands, altering and dropping of tables (use constraints while creating tables)

2. Performing DML commands like Insertion, Deletion, Modifying, Updating and selecting records based on conditions.

3. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, Constraints.

4. Implementation of Built in functions in RDBMS

5. Implementation of various aggregate functions in SQL

6. Implementation of Order By, Group By& having clause.

7. Implementation of set operators, nested queries and Join queries

8. Implement the usage of keys

9. Implement Queries with column alias and conversion functions

10. Implement Queries with sub string comparison

11. Implement SQL functions (String, Numeric, Date functions)

12. Implement relationship between tables

13. Implement various constraints.

14. Practice of SQL TCL commands like Rollback, Commit, Savepoint.

15. Practice of SQL DCL commands for granting and revoking user privileges.

16. Creation of Views and Assertions

17. Implementation of various control structures using PL/SQL

18. Creation of Procedures and Functions

19. Creation of Packages

20. Creation of database Triggers and Cursors

.

## REFERENCES

### LEARNING RESOURCES

**Text Books:**

1. ORACLE PL/SQL by example. Benjamin Rosenzweig, Elena Silvestrova, Pearson Education3rd Edition

2. ORACLE DATA BASE LOG PL/SQL Programming SCOTT URMAN, Tata Mc- Graw Hill.

3. SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande

| Semester II | Course Code: MCA20G21 | Credits: 1 |
|---|---|---|

## SEMINAR

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Acquire in-depth knowledge in specific area of study. |
| CO2 | Develop presentation skill and communication skill. |
| CO3 | Apply Professional skills for preparing presentation slides. |
| CO4 | Develop defending ability. |

### COURSE CONTENT

Seminar is an outstanding work, published in an international journal in the field that covered in the course need to be presented. The in depth knowledge of the underlying technology/method of the work is evaluated through this course. Students can make use of the presentation aids to deliver the theoretical aspects of the work. The interaction with the audience, Students and faculty is beneficial for the student to strengthen the different aspects of the presentation such as presentation skill, depth of knowledge, language and rendering, defending the questions.

| Semester III | Course Code: MCA20C31 | Credits: 3 |
|---|---|---|

## MACHINE INTELLIGENCE

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Investigate the applications of artificial intelligence |
| CO2 | Explain about learning systems and its application scope. |
| CO3 | Illustrate knowledge representation and its structures |
| CO4 | Identify the concepts of decision making theory. |
| CO5 | Explain the knowledge of advanced search and heuristic search techniques. |
| CO6 | Define machine learning |
| CO7 | Evaluate the performance of machine learning algorithms |
| C08 | Analyse the significance of machine learning for processing big data |

### COURSE CONTENT

**MODULE I:** Topics of Artificial Intelligence, Timelines of Artificial Intelligence, Branches of Artificial Intelligence, Applications of Artificial Intelligence. - Intelligent agents - structure, types of agents, environment, autonomous agents. Problem Solving - Production Systems, State space representation.

**MODULE II**: Knowledge Representation - Knowledge Management, Types of Knowledge, Knowledge representation, Knowledge base. - Knowledge Representation Structures - First Order logic, Unification algorithm, Frames, Conceptual Dependency, Scripts, Semantic network.

**MODULE III**: Reasoning - Types of reasoning, Non-monotonic reasoning, reasoning with Fuzzy logic, Rule Based reasoning, Case Based reasoning, Model based reasoning systems. – Bayes' rule, Bayesian networks, probabilistic inference, sample applications.

**MODULE IV**: Heuristic search techniques - Generate and test, Hill climbing, Simulated annealing, Problem reduction, AO* algorithm, Constraints satisfaction, Means - Ends analysis. Search Techniques - Graph search, Depth First Search, Breadth First Search, Iterative Deepening search, Uniform cost search, Greedy method, Best first search, Beam search, Branch and Bound search, A* algorithm.

**MODULE V:** Understanding Machine Learning -What is Machine Learning? - Defining Big Data- Big Data in Context with Machine Learning - Leveraging the Power of Machine Learning- Descriptive analytics - Predictive analytics.

## REFERENCES


- C. Bishop - “Pattern Recognition and Machine Learning”, Springer, 2007.
- K. Murphy - “Machine Learning: a Probabilistic Perspective”, MIT Press, 2012.
- Vinod Chandra S S, Anand H S - “Machine Learning: A Practitioners Approach”, Prentice Hall of India, New Delhi, 2020
- Vinod Chandra S S, Anand H S- "Artificial Intelligence and Machine Learning", Prentice Hall of India, New Delhi, 2014

| Semester III | Course Code: MCA20C32 | Credits: 3 |
|---|---|---|

## MOBILE APPLICATION DEVELOPMENT

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Able to work with Android development environment |
| CO2 | Identify the basic concepts in App development |
| CO3 | Critique mobile applications on their design pros and cons |
| CO4 | Create a simple mobile application |
| CO5 | Design an user interface using android framework |

### COURSE CONTENT

**MODULE I:** Introduction to android: the android platform, android sdk, eclipse installation, android installation, Building you first android application, understanding anatomy of android application, android Manifest file.

**MODULE II:** Android application design essentials: anatomy of an android applications, android terminologies, Application context, activities, services, intents, receiving and broadcasting intents, android Manifest file and its common settings, using intent filter, permissions.

**MODULE III :** Android user interface design essentials: user interface screen elements, designing user interfaces With layouts, drawing and working with animation.

**MODULE IV :** Testing android applications, publishing android application, using android preferences, managing application resources in a hierarchy, working with different types of resources.

**MODULE V:** Using common android apis: using android data and storage apis, managing data using sqlite, Sharing data between applications with content providers, using android networking apis, using Android web apis, using android telephony apis, deploying android application to the world.

## REFERENCES

## LEARNING RESOURCES

TEXT BOOKS:


1. T1. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education, 2nd ed. (2011)
2. Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt Ltd
3. Mark L Murphy, “Beginning Android”, Wiley India Pvt Ltd
4. Android Application Development All in one for Dummies by Barry Burd, Edition.


**Web Resources**

http://index-of.es/Android/Android_Application_Development_For_Dummies.pdf

| Semester III | Course Code: MCA20C33 | Credits: 3 |
|---|---|---|

## DATA SCIENCE WITH BIG DATA ANALYTICS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Illustrate the data science process |
| CO2 | Compare big data and data science |
| CO3 | Develop practical data analysis skills, which can be applied to practical problems. |
| CO4 | Provide a hands-on experience with real-world data analysis. |
| CO5 | Illustrate the working of Hadoop |
| CO6 | Compare the features of traditional databases and NoSQL databases |
| CO7 | Identify the Hadoop ecosystems |
| CO8 | Evaluate the data analysis techniques for applications handling large data |
| CO9 | Visualize and present the inference after performing the data analysis |

### COURSE CONTENT

**MODULE I:** Definition – Big Data and Data Science Hype – Why data science – The Current Landscape – Data Scientist - Data Science Process Overview – Defining goals – Retrieving data – Data preparation – Data exploration – Data modeling – Presentation.

**MODULE II:** Introduction to data visualization – Data visualization options – Filters – MapReduce – Dashboard development tools – Creating an interactive dashboard with dc.js-summary.

**MODULE III:** Data Science Ethics – Doing good data science – Owners of the data - Valuing different aspects of privacy - Getting informed consent - The Five Cs – Diversity – Inclusion – Future Trends.

**MODULE IV:** Introduction to Big Data: Characteristics of Data – Evolution of Big Data - Definition of Big Data - Challenges with Big Data - 3Vs of Big Data - Non Definitional traits of Big Data - Business Intelligence vs. Big Data - Data warehouse and Hadoop environment -Big Data Analytics: Classification of analytics

**MODULE V:** Terminologies in Big Data - CAP Theorem - BASE Concept. NoSQL: Types of Databases – Advantages – NewSQL - SQL vs. NOSQL vs NewSQL. Introduction to Hadoop: Features – Advantages – Versions - Hadoop Components – Architecture – HDFS - Hadoop vs. SQL – RDBMS vs. Hadoop – Hive- Pig

## REFERENCES

### LEARNING RESOURCES

### TEXT BOOKS


- Introducing Data Science, Davy Cielen, Arno D. B. Meysman, Mohamed Ali, Manning Publications Co., 1st edition, 2016
- An Introduction to Statistical Learning: with Applications in R, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer, 1st edition, 2013
- Tom White " Hadoop: The Definitive Guide" Third Edit on, O'reily Media, 2012.
- Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.


### ADDITIONAL REFERENCES


- Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly, 1st edition, 2015
- Doing Data Science, Straight Talk from the Frontline, Cathy O'Neil, Rachel Schutt, O' Reilly, 1st edition, 2013
- Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press (2013)

| Semester III | Course Code: MCA20P31 | Credits: 2 |
|---|---|---|

## LAB 1- MOBILE APPLICATION DEVELOPMENT

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Create a hello world mobile application |
| CO2 | Able to use the android framework for creating application |
| CO3 | Demonstrate UI design in the mobile application |
| CO4 | Identify the different components in Android application development environment |
| CO5 | Implement socket programming to make a connection from mobile phone |
| CO6 | Create an advanced mobile app for a given application |

### COURSE CONTENT

**List of experiments include**

- Android Java Basics: Getting started with Android development, project folder structure, simple programming, running project, generating build/APK of the app from Android Studio
- First application: Creating Android Project, Android Virtual Device Creation, Set up debugging environment, Workspace set up for development, Launching emulator, debugging on mobile devices.
- Basic UI design: Basics about Views, Layouts, Drawable Resources, Input controls, Input Events, Toasts.
- More UI Components: Layouts - GridView and ListView, Action bar, Adapters, Menus: Option menu, context menu, sub menu, Pickers - Date and Time, Spinners.
- Activity and Fragment: Activity, Fragment, Activity Lifecycle and Fragment Lifecycle.
- Intents: Implicit Intents, Explicit intents, communicating data among Activities.
- Navigation Drawer: Panel that displays the app's main navigation screens on the left edge of the screen
- Android Notifications – Toast, Dialogs (TimePicker, DatePicker, Progress, Alert), Notification Manager and Push Notification
- Write a sample program to show how to make a SOCKET Connection from Mobile phone.

### REFERENCES:

**LEARNING RESOURCES**


- Head first Android Development.
- Android Programming: Pushing the Limits, Wiley By Erik Hellman
- Android Application Development Black Book, Dreamtech Press, Pradeep Kothari, KLSI


**Web resources**

- https://www.iare.ac.in/sites/default/files/lab1/MAD%20LAB.pdf

| Semester III | Course Code: MCA20P32 | Credits: 2 |
|---|---|---|

## LAB-2 MACHINE LEARNING

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Use python to implement machine learning programs |
| CO2 | Implement classification and clustering tasks |
| CO3 | Implement dimensionality reduction techniques |

### COURSE CONTENT

List of experiments:-

1. **Classification**
   - Implement SVM , random forest, decision tree, K-NN classifier
2. **Clustering**
   - Implement K- Means algorithm
   - Dimensionality reduction techniques
   - Implement PCA, LDA.

| Semester III | Course Code: MCA20G31 | Credits: 1 |
|---|---|---|

## CASE STUDY (OOSE/ML/DS/Mobile App)

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Capture all of the details of the above mentioned courses which are relevant to the purpose of the study, within a real life context. |
| CO2 | Provide opportunity for active learning and reinforce the applicability of theory to practical situations. |
| CO3 | Learn the principles or the theory behind the concept through simulated problem solving and decision-making. |

### COURSE CONTENT

A case study is a detailed investigation done by a single individual or group on a specific topic that covers in the third semester. Case studies allow students to fully understand how an intervention worked, or why an intervention had an effect in a particular case.

| Semester IV | Course Code: MCA20C41 | Credits: 12 |
|---|---|---|

## PROJECT WORK

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Identify a problem statement for the final project. |
| CO2 | Perform literature review by analyzing the related works. |
| CO3 | Implement the existing work from the literature. |
| CO4 | Analyze the existing system capture the limitations. |
| CO5 | Propose a method improvement to overcome the limitations |
| CO6 | Evaluate and interpret the design and experimental results. |
| CO7 | Develop the skill set to write research papers and project thesis |

Major project work shall be done individually by each student under the guidance of a faculty member from the department. The Project proposals and synopsis submission shall be done in third semester itself. It is advisable to select the project topic and area keeping the following objectives in mind:

1. The project work shall give enough opportunity for the students to apply some of the skills and knowledge earned through the theory courses.
2. The student shall get an exposure in developing industry type applications/utility software for computer systems or mobile devices/in studying and analyzing theoretical concepts and presenting a comparative analysis of state-of-the-art techniques/in developing new or improved algorithms/in the use of soft computing techniques in selected area/discipline.

The students need to do the following activities:

1. The candidate shall submit a proposal for different projects before the assessment team. The team shall select and finalize one of the proposals. However if all proposals are not acceptable, candidate may be asked to submit new/modified proposals. The candidate shall prepare and submit a synopsis of the accepted proposal. A record of the accepted synopsis of each candidate shall be maintained in the department.
2. A detailed study of the requirements and feasibility of the proposed work shall be conducted by the candidate with the help of the project guide. A study phase report shall be presented before the assessment team within one month from the beginning of project work. The design of proposed work shall be completed and presented before the assessment team. The design shall be finalized with suggested corrections/updates.
3. The developed software/algorithm shall be implemented and demonstrated before the internal assessment team. A short presentation explaining the proposed work and experimental results shall also be made. The Project Report shall be finalized only after the internal presentation after correcting/updating the document based on the comments from internal assessment team.

| Semester IV | Course Code: MCA20C42 | Credits: 2 |
|---|---|---|

## COMPREHENSIVE COURSE VIVA

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Understand basic knowledge on different courses throughout the programme |
| CO2 | Evaluate the skill and knowledge acquired throughout the programme |

All students need to attend a course viva of the programme at the end of project work. All students will be evaluated by a panel of experts on their knowledge on different courses in the program, case studies done and the final project work. There will be evaluation of their professional development acquired by the programme.

| Semester IV | Course Code: MCA20G41 | Credits: 1 |
|---|---|---|

## MOOC 2

Each student has to successfully complete one MOOC course in the fourth semester from the topics related to (Communicative English, Research Methodology, Technical Writing, Entrepreneurship, Environmental studies, Cyber law or any new topic/technology introduced recently. etc.) with a minimum period of eight (08) weeks offered through E-learning platforms like SWAYAM, Coursera, etc. other than the course or related course done in Semester-1.

# ELECTIVE COURSES

# ELECTIVE – I

| Semester II | Course Code: MCA20E21 | Credits: 3 |
|---|---|---|

## PROBAILITY AND STATISTICS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Solve probability based problems |
| CO2 | Compute the binomial, Poisson, geometric probability distribution for a given data |
| CO3 | Perform Chi squared tests and write inference |
| CO4 | Identify the measures of dispersion and its significance |
| CO5 | Compute the coefficient of correlation from the given data and infer |
| CO6 | Estimate the Spearman's rank correlation coefficient |

### COURSE CONTENT

**MODULE I:** Random experiment and sample space, event, algebra of events. Definition of probability: properties of probability. Theorems on probability, conditional probability and independent events, laws of total probability, Baye's theorem and its applications.

**MODULE II:** Discrete and continuous random variables – Moments – Moment generating functions – Binomial, Poisson, Geometric, Uniform, Exponential and Normal distributions. Joint distributions – Marginal and conditional distributions - Central limit theorem.

**MODULE III:** Introduction to Statistics- Types of data: primary, secondary, quantitative and qualitative data. Types of Measurements: nominal, ordinal, discrete and continuous data. construction of frequency distributions for discrete and continuous data, Graphical representation of a frequency distribution by histogram, Chi-square and F distributions for mean, variance and proportion.

**MODULEIV** Measures of location or central tendency: Arithmetic mean, Median, Mode, Geometric mean, Harmonic mean. Measures of dispersion: Mean deviation, Quartile deviation, Standard deviation, Coefficient of variation. Moments: measures of skewness, Kurtosis.

**MODULE V** Correlation: Scatter plot, Karl Pearson coefficient of correlation, Spearman's rank correlation coefficient, Regression: Concept of errors, Principles of Least Square, Simple linear regression and its properties.

## REFERENCES

### LEARNING RESOURCES

### TEXTBOOKS


- Rohatgi V.K and Saleh E, An Introduction to Probability and Statistics, 3rd edition, John Wiley & Sons Inc., New Jersey, 2015.
- Gupta S.C and Kapoor V.K, Fundamentals of Mathematical Statistics, 11th edition, Sultan Chand & Sons, New Delhi, 2014.
- Mukhopadhyay P, Mathematical Statistics, Books and Allied (P) Ltd, Kolkata, 2015.
- Walpole R.E, Myers R.H, and Myers S.L, Probability and Statistics for Engineers and Scientists, Pearson, New Delhi, 2017.

| Semester II | Course Code: MCA20E22 | Credits: 3 |
|---|---|---|

## BLOCK CHAIN TECHNOLOGY

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Discuss and describe the history, technology, and applications of Blockchain |
| CO2 | Analyse the significance of crptocurrencies in the digital world |
| CO3 | Identify the functional/operational aspects of cryptocurrency ECOSCYSTEM |
| CO4 | Understand emerging abstract models for Blockchain Technology |
| CO5 | Illustrate the working of Ethereum Virtual Machine |
| CO6 | Assess Blockchain applications in a structured manner |
| CO7 | Analyse the process of creating a crypto currency |
| CO8 | Create an own Crypto token |

### COURSE CONTENT

**MODULE I:** Introduction to Blockchain: Evolution and Technology –Applications - Core components of Block Chain technology- Private block chain vs Public block chain - The consensus problem - Asynchronous Byzantine Agreement - AAP protocol and its analysis - Nakamoto Consensus on permission-less, nameless, peer-to-peer network

**MODULE II:** Abstract Models for BLOCKCHAIN - GARAY model - RLA Model - Proof of Work ( PoW) as random oracle - formal treatment of consistency, liveness and fairness - Proof of Stake ( PoS) based Chains - Hybrid models ( PoW + PoS)

**MODULE III:** cryptographic basics for cryptocurrency - a short overview of Hashing, signature schemes, encryption schemes and elliptic curve cryptography

**MODULE IV:** Bitcoin - Wallet - Blocks - Merkley Tree - hardness of mining - transaction verifiability - anonymity - forks - double spending - mathematical analysis of properties of Bitcoin.

**MODULE V:** Ethereum - Ethereum Virtual Machine ( EVM) - Wallets for Ethereum - Solidity - Smart Contracts - some attacks on smart contracts -(Trends and Topics) - Zero Knowledge proofs and protocols in Blockchain - Succinct non interactive argument for Knowledge ( SNARK) - pairing on Elliptic curves - Zcash.

## REFERENCES


- R.Pass et al, Fruitchain, a fair blockchain, PODC 2017 ( eprint.iacr.org/2016/916).
- R.Pass et al, Analysis of Blockchain protocol in Asynchronous networks , EUROCRYPT 2017, ( eprint.iacr.org/2016/454) . A significant progress and consolidation of several principles).
- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016.
- Blockchain by Melanie Swa, O'Reilly
- Mastering Bitcoin: Unlocking Digital Cryptocurrencies, by Andreas Antonopoulos
- Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits –

**On-line Sources**

- https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html
- https://www.hyperledger.org/projects/fabric

| Semester II | Course Code: MCA20E23 | Credits: 3 |
|---|---|---|

## PRINCIPLES OF MANAGEMENT

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Demonstrate managerial traits and become socially conscious managers. |
| CO2 | Explain the practices of business management of organizations |
| CO3 | Analyze and evaluate the productivity improvement mechanism |
| CO4 | Perform qualitative study on the need of good control system |
| CO5 | Evaluate the performance of an organization using various management principles. |

### COURSE CONTENT

**MODULE I :** Management- Meaning- Nature- Importance- Functions of Management- - Levels of Management- Managerial objectives-Managerial roles-School of Management thoughts: F.W Taylor, Henry Fayol, Peter Drucker.

**MODULE II:** Planning- Nature-Importance-Steps in planning- How to make effective planning Limitations- Types of plan- Decision making: Meaning- Types of decision- Factors involved in Decision Making.

**MODULE III:** Organization-Nature- Process of Organization-Importance of Organization-Principles of good Organization- Types of Organization: Line Organization- Functional Organisation- Line & Staff Organization- Committee Organization. Authority and Responsibility relationships: Centralization -Decentralization-Departmentation.

**MODULE IV:** Staffing: Systems Approach to staffing –Recruitment & Selection-Human Resource Development- Motivation: Meaning - Nature - Maslow's Need Theory- MC Gregor's X&Y Theories- Types of Motivation.
Direction- Meaning- Elements- Importance of Direction - Techniques of Direction -Span of Control Factors Determining Span of Control Qualities of a Good Supervision.

**MODULE V:** Leadership- Meaning- Nature- Need- Types of Leaders- Qualities of Leadership. Control Meaning- Control Process- Tools of Control- Characteristics of Good Control System-Co Ordination- Meaning- Determinants of Co-Ordination.
Information Technology &Management: E-business, E-commerce, Management Information Systems (MIS)-Types of Information systems.

## REFERENCES

**Text books**


1. DinkarPagare, “Principles of Management”, Sultan & Chand Publications, New Delhi
2. M.Govindarajan, S.Natarajan, ”Principles of Management”, PHI,New Delhi


**References**


1. L.M.Prasad, “Principles and Practice of Management”, Sultan & Chand Publications, New Delhi
2. P.C.Tripathi,N.Reddy, “Principles of Management”, Sultan & Chand Publications, New Delhi
3. S.S Khanka ,”Organisational Behavior” , S.Chand& Company Ltd
4. Koontz &Wheinrich ,”Essentials of Management “, PHI Publications
5. Griffin, Ricky W, ”Organisational Behaviour”, Houghton Mifflin co., Boston

| Semester II | Course Code: MCA20E24 | Credits: 3 |
|---|---|---|

## CLOUD COMPUTING

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Understand the basic concept of Cloud Computing |
| CO2 | Interpret the various types of Cloud services |
| CO3 | Illustrate the use of various cloud services available online |
| CO4 | Identify the significance of implementing virtualization techniques |
| CO5 | Understand the need of security mechanisms in cloud. |

### COURSE CONTENT

**MODULE I :** Cloud Computing Fundamentals: History of Cloud Computing, Cloud Computing definition, Types of cloud, Cloud services: Advantages and disadvantages of cloud computing, Evolution of Cloud Computing , Applications of cloud computing, Business models around Cloud – Major Players in Cloud Computing

**MODULE II:** Cloud Services and File System: Types of Cloud services: Software as a Service - Platform as a Service – Infrastructure as a Service - Database as a Service- Monitoring as a Service – Communication as services. Service providers- Google App Engine, Amazon EC2, Microsoft Azure, Sales force.

**MODULE III:** Collaborating Using Cloud Services: Email Communication over the Cloud - CRM Management - Project Management-Event Management - Task Management – Calendar - Schedules - Word Processing – Presentation – Spreadsheet - Databases – Desktop - Social Networks and Groupware.

**MODULE IV :** Virtualization: Basics of Virtualization - Types of Virtualization - Implementation Levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU, Memory, I/O Devices - Virtual Clusters and Resource management – Virtualization for Data-center Automation. Introduction to MapReduce, GFS, HDFS, Hadoop Framework.

**MODULE V :**Security in the Cloud: Security Overview – Cloud Security Challenges and Risks – Software-as-a-Service Security – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security - Identity Management and Access Control – Autonomic Security.

## REFERENCES

## LEARNING RESOURCES

### References


- Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing ”A Practical Approach” McGraw□Hill.
- Lovely Professional University, Cloud Computing, Excel Book Private Ltd.
- Virtualization, http://stmarysguntur.com/wp-content/uploads/2018/02/PDFsam_merge.pdf.
- Ronald L. Krutz, Russell Dean Vines, “Cloud Security – A comprehensive Guide to Secure Cloud Computing”, Wiley – India, 2010.
- Kai Hwang, Geoffrey C Fox, Jack G Dongarra, “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012.
- John W.Rittinghouse and James F.Ransome, “Cloud Computing: Implementation, Management, and Security”, CRC Press, 2010.
- Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing, A Practical Approach”, TMH, 2009.
- Kumar Saurabh, “ Cloud Computing – insights into New -Era Infrastructure”, Wiley India,2011.

## ELECTIVE II

| Semester III | Course Code: MCA20E31 | Credits: 3 |
|---|---|---|

### ADVANCED MACHINE LEARNING MODELS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Compare different quantification methods of classification |
| CO2 | Differentiate different clustering techniques and algorithms |
| CO3 | Implement Support Vector Machine algorithm and its variants |
| CO4 | Explain different learning algorithms based on decision tree |
| CO5 | Study different ANN techniques and deep techniques |
| CO6 | Compare different ANN networks and working structure |
| CO7 | Compare different Deep architectures and their learning models |

### COURSE CONTENT

**MODULE I:** Learning - Types of learning, Learning of Input/ Output Function, history and timelines of machine learning, Aspects of machine learning, Machine Learning Applications and examples, intelligent agents. Quantification of classification - Threshold Fixing, ROC Graphics, ROC formulation. Supervised vs. Unsupervised learning - Prediction system, Training, testing and validation datasets, cross validation. Supervised learning model - Bias-variance trade-off, classification problems.

**MODULE II:** Unsupervised learning model - clustering, data compression, PCA. Clustering - k-Means clustering, Facts about k-means, k-Means clustering weakness. Fuzzy clustering, hierarchical clustering Agglomerative and Divisive Clustering, Hierarchical Agglomerative Clustering, Cluster similarity.

**MODULE III:** Support Vector Machines- Margins, Learning a maximum hyperplane, Kernel functions, Linear SVM, Non-linear SVM, Applications of SVM. Decision Trees - Decision tree construction, types of decision trees. Decision tree algorithms - C4.5 algorithms, ID3 algorithm,

CART, random forest. Univariate trees and Multivariate trees - functional tree, J48 tree,J48-graft, Best-first trees, Naive Bayesian tree.

**MODULE IV:**ANN basics, Types of networks, The perceptron, RBF networks, Self-organising maps, Adaptive resonance theory, Recurrent neural network, Hopfield networks, Boltzmann machines, Probabilistic neural network

**MODULE V:**Deep architecture -Recurrent and Recursive networks, Bidirectional RNNs, Deep Recurrent Networks, Recursive Neural Networks, LSTM, GRU. Image captioning, word prediction. Deep Belief networks, Convolutional neural networks, Deep reinforcement learning, Geometric stability, Applications of deep learning.

## REFERENCES

### LEARNING RESOURCES


- Aggarwal Charu – “Neural Networks and Deep Learning”, Springer, 2015.
- AurélienGéron’s, "Hands-On Machine Learning with Scikit-Learn and TensorFlow", O'Reilly Media, Inc.,2017.
- Ian Goodfellow, YoshuaBengio, Aaron Courville – “Deep Learning”, MIT Press, 2016.
- Mike Krebbs - "Deep Learning with Python", CreateSpace Independent Publishing Platform, 2018.
- Vinod Chandra S S, Anand H S - “Machine Learning: A Practitioners Approach”, Prentice Hall of India, New Delhi, 2020.

| Semester III | Course Code: MCA20E32 | Credits: 3 |
|---|---|---|

## BIOLOGICALLY INSPIRED ALGORITHMS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Explain bio inspired computing fundamentals |
| CO2 | Classify optimization problems and its types |
| CO3 | Illustrate Genetic algorithm and its applications |
| CO4 | Compare different Ant Colony Optimization algorithmic models |
| CO5 | Compare different Artificial Bee Colony Optimization algorithmic models |
| CO6 | Illustrate Particle swam optimization algorithm with an example |
| CO7 | Discuss different natural inspired computing algorithms |

### COURSE CONTENT

**MODULE I:** Models of Life and Intelligence - Fundamentals of bio-inspired models and bio-inspired computing. Evolutionary models and techniques, Swarm models and its self-organisation, swarm and evolutionary algorithms. Optimisation problems – single and multi-objective optimisation, heuristic, meta-heuristic and hyper heuristic functions.

**MODULE II:** Genetic algorithms - Mathematical foundation, Genetic problem solving, cross over and mutation. genetic algorithms and Markov process, applications of genetic algorithms. Other evolutionary algorithms.

**MODULE III:** Ant Colony Algorithms - Ant colony basics, hybrid ant system, ACO in combinatorial optimisation, variations of ACO, case studies.

**MODULE IV:** Particle Swam algorithms - particles moves, particle swarm optimisation, variable length PSO, applications of PSO, case studies. Artificial Bee Colony algorithms - ABC basics, ABC in optimisation, Multi-dimensional bee colony algorithms, applications of bee algorithms, case studies.

**MODULE V:** Selected nature inspired techniques - Cuckoo search, Firey algorithm, SDA algorithm, bat algorithm, Social spider algorithm, Glow-worm Swarm optimization, Plant growth

adaptation in optimization, Termite colony optimization, Intelligent water drops algorithm, Artificial immune system, Flower pollination algorithm, case studies.

## REFERENCES

### LEARNING RESOURCES


- Albert Y.Zomaya - "Handbook of Nature-Inspired and Innovative Computing", Springer, 2006
- Floreano, D. and C. Mattiussi -"Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies", MIT Press, 2008
- Marco Dorrigo, Thomas Stutzle -" Ant Colony Optimization", Prentice Hall of India, New Delhi, 2005
- Vinod Chandra S S, Anand H S - "Machine Learning: A Practitioners Approach", Prentice Hall of India, New Delhi, 2020

| Semester III | Course Code: MCA20E33 | Credits: 3 |
|---|---|---|

## BIOINFORMATICS

### COURSE OUTCOMES

| | |
|---|---|
| **CO1** | Demonstrate the basic knowledge in Life science |
| **CO2** | Compare RNA and DNA |
| **CO3** | Analyse the DNA sequence and perform matching |
| **CO4** | Illustrate the working of pair wise alignment method |
| **CO5** | Identify the bioinformatics databases |
| **CO6** | Perform sequence alignment in DNA sequences |
| **CO7** | Understand the significance of Bioinformatics as a multidisciplinary research area |

### COURSE CONTENT

**MODULE I Introduction to life Science :** Characteristics of life, Levels of biological Organization, cell as basic MODULE of life, cell theory, structure of Prokaryotic cell and Eukaryotic cell, Primary and secondary structure of DNA, Chargaff's Rules, Different forms of DNA, RNA, structural organization of DNA, Gene and genetic information

**MODULE II: Bioinformatics**: History of Bioinformatics, Definition of Bioinformatics, Bioinformatics versus Computational Biology, Goals of Bioinformatics analysis, Bioinformatics technical tool box, Biological data, File format, conversion of file format, Data retrieval system, Genome browsers.

**MODULE III:** Analyzing DNA sequence, IUPAC code for DNA sequence, ORF, palindromes in DNA sequence, RNA sequence analysis; FASTA format. Sequence analysis/Alignment: DNA sequence, RNA sequence, Protein sequence, sequence alignment classifications, Scoring Matrices – PAM, BLOSUM; Sequence Alignment: Introduction to Sequence Comparison - Pairwise Alignment Method (DOT PLOT method) and Multiple Analyses of Protein Structures

**MODULE IV:** Databases: Bioinformatics databases, Types of databases, Nucleotide sequence databases, Primary nucleotide sequence databases-EMBL, Gene Bank, DDBJ; Secondary nucleotide databases, Protein sequence databases- SwissProt/TrEMBL, Protein structure databases- Protein Data Bank

**MODULE V:** Human Genome Project, Importance of Perl language in Bioinformatics, Applications of Bioinformatics in Biodiversity, Human Genetics, Gene Therapy, Agriculture, Computer-Aided Drug Design, DNA Fingerprinting.

## REFERENCES

### Text books


1. P S Verma, V K Agarwal, Cell Biology, enetics, Molecular Biology, Evolution and Ecology, S. Chand Publications.
2. S C Rastogi, N Mendiratta, P Rastogi, Bioinformatics Methods and Applications, PHI
3. Jin Xiong, Essential Bioinformatics, Cambridge University Press
4. Jean-Michel Claverie, Cedric Notredame, Bioinformatics: A Beginner's Guide, Wiley, 2006
5. Dr. K Mani & N Vijayaraj, Bioinformatics: A practical approach, Aparna Publications


### Additional and Web -Resources

1. https://nptel.ac.in/courses/102/106/102106065

| Semester III | Course Code: MCA20E34 | Credits: 3 |
|---|---|---|

## CYBER FORENSICS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Identify, collect, preserve and analyze data in a way that preserves the integrity of the evidence collected so it can be used effectively in a legal case |
| CO2 | Define and discuss the concepts of computer forensics |
| CO3 | Able to apply the concepts of computer investigations |
| CO4 | Identify and apply current practices for processing crime and incident scenes |
| CO5 | Demonstrate the boot processes and disk structures of various operating system environments |
| CO6 | Perform e-mail investigations |
| CO7 | Conduct basic computer forensic analysis |
| CO8 | Identify and apply current practices for data discovery recovery and acquisition |
| CO9 | Select and apply current computer forensics tools |

### COURSE CONTENT

**MODULE-I:** Introduction to Computer Forensics:- History of computer forensics, Developing computer forensics resources, Preparing for computer investigations, Understanding law enforcement agency investigations, corporate investigations, maintaining professional conduct, understanding computer investigations- preparing, taking a systematic approach, procedures for corporate high tech investigations, Understanding data recovery workstations and software, Conducting an investigation, completing the case, requirements for forensic lab certification, determining the physical requirements for a computer forensics lab, Selecting a basic forensic workstation, building a business case for developing a forensic lab

**MODULE-II:** Data Acquisition:- Storage formats for digital evidence, Determining the best acquisition method, contingency planning for image acquisitions, using acquisition tools, validating data acquisitions, performing RAID data acquisitions, using remote network acquisition tools, using other forensic acquisition tools, processing crime and incident scene-identifying digital evidence, collecting evidence in private sector incident scenes, processing law enforcement crime scenes, preparing for a search, securing a computer incident or crime scene-seizing digital evidence at the scene, storing digital evidence

**MODULE-III:** Working with windows and DOS systems-file systems: exploring Microsoft file structures, examining NTFS disks, whole disk encryption, the windows registry, Microsoft and Ms-DOS start up tasks, virtual machines, Evaluating computer forensics tools needs, computer forensics software and hardware tools, validating and testing forensics software, the Macintosh file structure and boot process, examining UNIX and LINUX disk structures and boot processes, examining CD data structures, examining SCSI Disk, examining IDE/EIDE & SATA devices

**MODULE-IV:** Analysis and Validation:- Determining what data to collect and analyze, validating forensic data, addressing data- hiding techniques, performing remote acquisitions. Recovering Graphics Files-Recognizing, locating and recovering graphic files, understanding data compression, copy rights issues with graphics, identifying unknown file formats, copy right issues with graphics. Network Forensics- overview, performing live acquisitions, developing standard procedures for network forensics, using network tools.

**MODULE-V:** E-Mail investigations:- Role of E-mail in investigations, exploring the roles of the client and server, investigating e-mail crimes and violations, understanding E-mail servers, specialized e-mail forensic tools. Report writing for high tech investigations- importance of reports, guidelines for writing, generating report findings with forensic software tools.

**Textbooks:**


1. Bill Nelson, Amelia Philips, Frank Enfinger, Christofer Steuart‘ Guide to Computer Forensics and Investigations ‘, Second Indian Reprint 2009, Cengage Learning India Private Limited
2. Eoghan Casey, ‘Digital Evidence and Computer Crime’ Edition 3, Academic Press,2011
3. Marjie Britz,’ Computer Forensics and Cyber Crime: An Introduction’ Edition 2, Prentice Hall, 2008


## REFERENCES


1. Practical guide to Computer Forensics-David Benton and Frank Grindstaff, 2006, Book of Surge Publishing, 2006
2. Christopher L.T Brown Charles,’ Computer Evidence: Collection & Preservation ‘-, River Media publishing ,Edition1, 2005
3. Keith Jejune, Richard Bejtlich and Curtis W. Rose,’ Real Digital Forensics ‘Addison-Wesley publishers, 2005

## ELECTIVE III

| Semester III | Course Code: MCA20E35 | Credits: 3 |
|---|---|---|

### SOCIAL NETWORK ANALYSIS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Introduce basic concepts, evolution and related applications of social network analysis. |
| CO2 | Mathematical representation of social networks. |
| CO3 | Web based social networks, Data extraction from web based communities |
| CO4 | Network metrics, network effects and network models |
| CO5 | Extraction and mining communities in web social networks |
| CO6 | Application of software tools for social network analysis |

### COURSE CONTENT

**MODULE I :** Introduction to Social Network Analysis(SNA): Characteristic features of social networks, Key concepts, development, and applications of social network analysis. Electronic sources for network analysis: Electronic discussion networks, Blogs and online communities – applications in web-based networks; Mathematical representation of social networks: Notation for social network data, graph theoretic notation, Sociometric Notation, Algebraic Notation, Graphs and Matrices.

**MODULE II:** Measures of centrality:- Degree centrality, Betweenness centrality, Closeness centrality, Eigenvector centrality, Page rank, Transitivity and Reciprocity. Computing network metrics using software tool Gephi and R language.

**MODULE III:** Network Effects, Power Laws and Rich Get Richer Phenomena, Scale free networks, The Small-World Phenomenon, Cascading Behavior in Networks, Models and Methods in Social Network Analysis: Random Graph Model, Small World Model, Preferential Attachment Model.

**MODULE IV:** Diffusion on Networks: Information Diffusion and Cascade Model, Epidemics – SIR & SIS Model, Threshold Model, Simple and complex contagion , Detecting communities in social networks - Definition of community - Evaluating communities - Methods for community detection and mining - Applications of community mining algorithms - Tools for detecting communities social network infrastructures and communities

**MODULE V:** Application of Data Mining in Social Networks: Overview on Data Mining and its various techniques, Overview on web mining

## REFERENCES

**Text books**


- David Easley and Jon Kleinberg, Networks, Crowds, and Markets: Reasoning about a Highly Connected World, Cambridge University Press, 2010 2.
- Zafarani, Abbasi and Liu, Social Media Mining: An Introduction, Cambridge University Press, 2014
- Newman MEJ Netowrks, An Introduction.Oxford University Press 2010
- Dion Goh and Schubert Foo, "Social information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively", IGI Global Snippet, 2008.
- Max Chevalier, Christine Julien and Chantal Soulé-Dupuy, "Collaborative and Social Information Retrieval and Access: Techniques for Improved user Modelling", IGI Global Snippet, 2009.
- John G. Breslin, Alexandre Passant and Stefan Decker, "The Social Semantic Web", Springer, 2009.
- GuandongXu ,Yanchun Zhang and Lin Li, "Web Mining and Social Networking – Techniques and applications", First Edition Springer, 2011.


**Additional and Web –Resources**


- https://gephi.org/users
- derekgreene.com/slides/derekgreene_gephi_slides.pdf
- https://www.worldscientific.com/doi/abs/10.1142/9789814277327_0010
- https://cran.r-project.org/web/packages/igraph/igraph.pdf

| Semester III | Course Code: MCA20E36 | Credits: 3 |
|---|---|---|

## DIGITAL MARKETING

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Use digital marketing for performing multiple goals within a larger marketing and/or media strategy |
| CO2 | Identify the major digital marketing channels - online advertising: Digital display, video, mobile, search engine, and social media |
| CO3 | Develop, evaluate, and execute a comprehensive digital marketing strategy and plan |
| CO4 | Explain the process of search engine optimization |
| CO5 | Illustrate the working of a CRM model |
| CO6 | Able to develop a swot analysis and to define a target group. |
| CO7 | Provide the business opportunities of Instagram citing the real world examples |

### COURSE CONTENT

**MODULE I:** Introduction to digital marketing - Digital vs. Real Marketing- Understanding Marketing Process -Understanding Digital Marketing Process, Increasing Visibility, Types of visibility, Examples of visibility, Visitors Engagement, Examples of engagement -Bringing Targeted Traffic

**MODULE II:** 5 D's in digital marketing- Digital Marketing Channels- Creating initial digital marketing plan - Content management- SWOT analysis -Target group analysis

**MODULE III:** Web design - Optimization of Web sites Search engine optimization -- search engines –working of search engines- Major functions of a search engine- keyword- Different types of keywords- MS Expression Web- SEO Optimization - Writing the SEO content – Search engine marketing- SEM

**MODULE IV:** Introduction to CRM - CRM platform - CRM models- Introduction to Web analytics - Web analytics - levels - Introduction of Social Media Marketing- Creating a Facebook page - Visual identity of a Facebook page - Types of publications

**MODULE V:** Business opportunities and Instagram options • Optimization of Instagram profiles • Integrating Instagram with a Web Site and other social networks • Keeping up with posts, Trends in digital advertising

## REFERENCES

### LEARNING RESOURCES

### TEXTBOOK


- Ryan, D. (2014 ). Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation, Kogan Page Limited.
- The Beginner's Guide to Digital Marketing (2015). Digital Marketer. Pulizzi,J.(2014) Epic Content Marketing, Mcgraw Hill Education.

| Semester III | Course Code: MCA20E37 | Credits: 3 |
|---|---|---|

## INTERNET OF THINGS

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Able to understand the characteristics & Architecture of IoT |
| CO2 | Explore the Physical & Logical Design of IoT and Domain Specific IoTs & Various blocks of IoT |
| CO3 | Demonstrate the difference between M2M and IoT |
| CO4 | Able to learn Sensors & Actuators. Roles of Sensors and Actuarors |
| CO5 | Illustrate the working of IoT reference model & architecture. |
| CO6 | Discuss various IoT protocols |
| CO7 | Investigate the Domain Specific application of IOT & various challenges of IoT |

### COURSE CONTENT

**MODULE I:** Introduction and Concepts of IOT : Introduction to IOT, Definition and characteristics of IOT, Architecture of Internet of Things, Components of IOT, Physical and logical design of IOT, IOT enabling technologies, IOT levels and deployment templates, Domain specific IOTs, Design and Functional blocks of IoT, Communication models & APIs

**MODULE II:** IOT & Machine to Machine, Difference between IoT and M2M, Software defined Network; Technologies – Sensors Roles of sensors in IOT. Actuators- Role of actuators, Gateways, Local & Global Connectivity, Communication models & APIs

**MODULE III:** Reference Model and architecture- IoT reference Model - IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints- Introduction, Technical Design constraints-hardware, Data representation and visualization, Interaction and remote control.

**MODULE IV:** IoT Protocols : PHY/MAC Layer - 3GPP MTC, IEEE 802.11, IEEE 802.15 ,Data Link – Z Wave, BLE, ZigBee Network Layer- 6LoWPAN,RPL, CORPL, CARP , Transport Layer TCP, MPTCP, UDP, DCCP, SCTP) – Session Layer- CoAP, XMPP, AMQP, MQTT –Security in IoT Protocols – MAC 802.15.4 , 6LoWPAN, RPL, Application Layer Protocols

**MODULE V:** Domain specific applications of IoT Home automation, Industry applications, Surveillance applications, Other IoT applications; Web of Things to the Cloud of Things. – Integration of Cloud & IoT .Challenges in IoT - Design challenges, Development challenges, Security challenges, other challenges, Characteristics and challenges of Internet of Vehicles (IoV).

## LEARNING RESOURCES

### TEXT BOOK


1. Arshdeep Bahga and Vijay Madisetti , “ Internet of Things: A Hands-on Approach “
2. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnourskos, David Boyle ,”From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”


### REFERENCES


1. Rajkumar Buyya ,Amir Vahid Dastjerdi, Internet of Things Principles and Paradigms
2. Pethuru Raj and Anupama C. Raman ,”The Internet of Things: Enabling Technologies, Platforms, and Use Cases"
3. David Hanes, “ IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things”, Cisco Press, Pearson


### Web Resources

- https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/#sec4
- https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_dlc.pdf
- https://onlinecourses.nptel.ac.in/noc19_cs65/preview

| Semester III | Course Code: MCA20E38 | Credits: 3 |
|---|---|---|

## CYBER SECURITY & CRYPTOGRAPHY

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Discuss the usage of Mathematical concepts and Number theory in design of cryptographic algorithms |
| CO2 | Identify the measures for data protection by ensuring confidentiality, integrity, authentication and non-repudiation |
| CO3 | Examine the strength of cryptographic algorithm by crypt analysis. |
| CO4 | Demonstrate the security services that can be implemented with the methods of modern cryptography. |
| CO5 | Explain the Message Authentication methods |
| CO6 | Illustrate the working of Asymmetric Key Encryption techniques. |
| CO7 | Explain security protocols for protecting data on networks |
| CO8 | Analyze various authentication and security protocols such as SSL,TLS IPSec etc. |
| CO9 | Comprehend the usage of firewalls and Intrusion Detection Systems for securing data |

### COURSE CONTENT

**MODULE 1 :** Mathematical Concepts of Cryptography – Divisibility and Division Algorithm – Euclidean Algorithm, Modular Arithmetic- Groups - Rings – Fields, FiniteFieldsoftheFormGF(p)–PolynomialArithmetic–FiniteFieldsoftheForm Introduction to Number Theory – Prime Numbers – Fermat's and Euler's Theorems, Testing for Primality– Discrete Logarithms. Case Study: Implement Encryption using binary Exclusive OR (XOR)

**MODULE 2:** Security Trends: Security attacks- Security services- Security Mechanisms, Classical Encryption Techniques -Symmetric Cipher Model – Substitution Techniques – Transposition Techniques – Rotor Machines- Steganography. Block Cipher Principles, DES, Differential and Linear Cryptanalysis, Block Cipher Design Principles, Multiple encryption and Triple DES, Block Cipher Modes of Operation, Advanced Encryption Standard, Blowfish Algorithms

**MODULE 3 :** Principles of Public Key Cryptosystems, RSA Algorithm, Key Management, Message Authentication and Hash Functions, Authentication Requirements, Authentication Functions, Message Authentication, Hash Functions, Security of Hash Functions and MACs, Digital. Signatures, Authentication Protocols, Digital Signature Standard.

**MODULE 4 :** Network Security Applications: Kerberos, X.509 Authentication Service, Public Key Infrastructure, Pretty Good Privacy, S/MIME, IP Security Overview, IP Security architecture, Authentication Header, Encapsulating Security Payload, Combining Security associations, Key Management.

**MODULE 5:** System Security: Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction, Intruders, Intrusion Detection, Password Management, Malicious Software, Firewalls, Trusted Systems.

**LEARNING RESOURCES**

**TEXT BOOK**


1. William Stallings, "Cryptography and Network Security-Principles and Practices", Fifth Edition, Pearson Education
2. Atul Kahate ,"Cryptography and Network Security, 3rd Edition


## REFERENCES


1. Behrouz A Forouzan, "Cryptography and Network Security", Tata McGraw Hill, 2008
2. Matt Bishop, "Computer Security: Art and Science", Addison-Wesley Professional, 2003
3. Wade Trappe, Lawrence C Washington, "Introduction to Cryptography with Coding Theory", Second Edition, Pearson Education

## BRIDGE COURSES

| Semester I | Course Code: MCA20B01 | Credits: 3 |
|---|---|---|

### PRINCIPLES OF PROGRAMMING

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Awareness about programming paradigms, programming environments ,need of debugging |
| CO2 | Overview of functional programming languages and its evolution |
| CO3 | Overview of Logical programming languages and its evolution |
| CO4 | Introduction to syntax and semantics. |
| CO5 | Awareness about Types of Grammars and parsing |
| CO6 | Introduction to Scripting langauges |
| CO7 | Introduction to Object oriented programming language, OOP concepts using different languages |

### COURSE CONTENT

**MODULE I**: Introduction to Programming Paradigms, Characteristics of Programming Languages, Programming language paradigms: Imperative, Object-Oriented, Functional, Logic, Event Driven and Concurrent Programming, Programming environments: Compilers, Interpreters, Interactive development tools, Debugging tools.

MODULE II: Functional Programming languages – LISP, Language overview, processes in functional programming, evaluation. Two descendants of LISP- SCHEME Language, Common LISP .Meta Language (ML), Haskell Language (basics only).Overview and Evaluation of ALGOL 60, COBOL, BASIC, PL/I. Early Dynamic languages-APL, SNOBOL, Origin and Characteristics. Beginning of Data Abstraction – SIMULA, overview. Early Descendents of ALGOLs-PASCAL, C, Overview and Evaluation.

MODULE III: Logical Programming language – PROLOG, Ada, Overview and Evaluation. Extended Logic Programming, Object Oriented Programming – Smalltalk, Language overview and Evaluation. Combining Imperative and Object Oriented features-C++, Overview, evaluation, Related languages-Objective C, Delphi, Go (Basics only). Imperative Based Object Oriented language –Java, Overview, Evaluation. Scripting languages-Origin and characteristics of Perl, JavaScript, PHP, Python ..NET languages-C#, Overview, Evaluation. Mark-up/Hybrid Languages-XSLT, JSP (Basics only).

**MODULE IV:** Describing Syntax and semantics-Language Recognizers, Language Generators, Grammars and Recognizers-Attribute Grammars, Dynamic semantics, Axiomatic Semantics. Lexical and syntax analysis-Parsing , Top down and bottom up parsers(Basics only).Names ,Bindings and scopes-Names, variables, Concept of binding-static and dynamic, scopes-static and dynamic.

**MODULE V:** Scripting Language- key Concepts, Imperative programming languages-variables, type checking, scope, data type, arithmetic expressions, control flow, sub programs. Object Oriented Programming languages-C++ and its support for OOP, Java and OOPS,C # and OOPS, Python and OOPS, Event handling in Java.

## REFERENCES

**Text books**


- R. Sebesta, Concepts of Programming Languages, Addison Wesley
- Principles of Programming languages – A Paradigm Approach – Syed Buhari, McGrawHill,
- Debashish Jana, Java and Object-Oriented Programming Paradigm, Prentice Hall., 2008.
- Van Roy, Haridi, Concepts, Techniques and Models of Computer Programming, MIT Press,2004.
- Robert Lafore, Object-Oriented Programming with C++
- Bruce Eckel , Thinking in Java
- J. Reynolds, Theories of Programming Languages, Cabridge University Press.
- Duckett Jon, Beginning Web Programming with HTML, XHTML and CSS, Wrox, 2004.

| Semester I | Course Code: MCA20B02 | Credits: 3 |
|---|---|---|

# DIGITAL LOGIC AND COMPUTER ARCHITECTURE

## COURSE OUTCOMES

| | |
|---|---|
| CO1 | Able to understand of Digital logic fundamentals and computer architecture |
| CO2 | Identify the organization of computer memory and peripherals |
| CO3 | Explain the data transfer and control mechanisms in digital computers |
| CO4 | Summarise the computer arithmetic and processor organization |
| CO5 | Evaluate the performance of processors through parallel processing |

## COURSE CONTENT

**MODULEI:** Digital computers, Logicgates, Combinational logic , circuits:Adders,Subtractors,Encoders,Decoders,Multiplexers,Demultiplexers.Sequential circuits Flipflops, Counters, Shift Registers, Number representation and arithmetic operations, Digitalcodes, Error detection and correction.

**MODULE II:** Memory organization: Memory hierarchy – Main memory – Auxiliary memory – Associative memory – Cache memory – Virtual memory, Input-output organization: Peripheral devices – I/O interface – Asynchronous data transfer-DMA-Input/output processor(IOP)

**MODULE III:**CPU: Register and stack organization – Instruction formats – Addressing modes – Data transfer and manipulation – Program control – RISC Control Unit: Control Signals-Control memory-Hardwired Control -Microprogrammed Control

**MODULE IV:** Pipelining – Arithmetic and instruction pipeline – RISC pipeline –vector processing- Array Processors, Parallel Processing and Performance: Hardware Multithreading

**MODULE V:** Vector (SIMD) Processing - Graphics Processing Units, Shared-Memory Multiprocessors, Cache Coherence - Write-Through Protocol, Write-Back protocol, Snoopy Caches, Directory-Based Cache Coherence , Message-Passing Multicomputer.

## LEARNING RESOURCES

**Text books:**

1. M. Morris Mano, “Computer System Architecture”, Pearson Education .

2. John P Hayes, Computer Architecture and Organization, McGraw-Hill Book Company.

3. Thomas.L.Floyd, ”Digital Fundamentals”, Pearson Education.

## REFERENCES

1. Computer organization And Embedded Systems, Hamacher, Vranesic, Zaky,Manjikian, McGraw-Hill.

2. Manish Saraswat, ‘Computer Architecture And Organisation’, Vayu Education Of India.

3. Tanenbaum A.S, ‘Structured Computer Organization’, Prentice Hall of India

4. Malvino A. P. and Donald P. Leach, “Digital Principles and Applications”, McGraw Hill Publications.

| Semester I | Course Code: MCA20B03 | Credits: 2 |
|---|---|---|

## PROGRAMMING C LAB

### COURSE OUTCOMES

| | |
|---|---|
| CO1 | Awareness about Compilation and Execution of programs, Pre-processor commands, syntax, data types, storage class, built in functions in C, Working with Formatted Input and Output. |
| CO2 | Develop program skills to implement different operators in C–Arithmetic, Logical, Assignment, bit wise, sizeof () and conditional operator. |
| CO3 | Develop program skills for decision making- using if, if…else, nested if, Conditional operator, switch, and skills for using loop structures, nested loops, and loop control statements-break, continue |
| CO4 | Develop C program for defining and calling function, Concept of call by value, call by reference, Demonstrate the difference between iteration and recursion in terms ofC programming |
| CO5 | Implementation of arrays, structures & pointers |
| CO6 | String handling and File Management concepts |

### COURSE CONTENT

**MODULE I :** Introduction to Pre-processor commands, syntax, comments, built in functions, Working with formatted input and output. Small programs implementing basic data types – integer, float, char, variables and constants. Program using storage class –static; Operators in C - arithmetic operators, Logical operators-AND, OR, NOT, assignment operators bit wise operators, conditional operator and sizeof() operator.

**MODULE II:** Decision making in C using -if, if …else, nested if statements. switch…case statement, entry and exit control loops structures-for loop, while loop, do…while loop; Nested loops; Loop control statements – break, continue and goto.

**MODULE III:** Functions in C- built in and user defined functions-Local and global variables; actual and formal parameters; simple example of declaring and using functions with (i) no argument, no return (ii) with argument, no return,(iii) no argument, with return,(iv) with argument, with return; function calls, call by value and call by reference; function recursion.

**MODULE IV:** Arrays in C – One dimensional and two dimensional arrays. Implement pointers- pointers to arrays and array of pointers - (Programs to demonstrate implementation of 1 D and 2 D arrays using pointers).

**MODULE V:** Concept of header files, C program to implement basic string and mathematical functions, Structures -accessing structure members, pointer to structure, File handling programs in C – open, read and write operations in file.

### REFERENCES

**Text books**


- Ashok N.Kamthene, Programming in C, Pearson Education, Third edition.


**Additional and Web –Resources**


- E Balaguruswamy , Programming in ANSI C, Mc Graw hill, Eighth Edition.