# Indira Gandhi Delhi Technical University for Women

(Established by Govt. of Delhi vide Act 09 of 2012)

Kashmere Gate, Delhi–110006

# Scheme of Examination

# &

# Detailed Syllabus

(w.e.f. Academic Year 2013-2014 onwards)

for

# Master of Computer Applications

## Department of Information Technology

## PROGRAMME OUTCOMES

Post Graduates of Master of Computer Application will be able to:

**PO1**. Apply knowledge of Computing fundamentals, Computing specialization, Mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.

**PO2**. Design and develop applications to analyze and solve all computer science related problems.

**PO3**. Design applications for any desired needs with appropriate considerations for any specific need on societal and environmental aspects.

**PO4**. Analyze and review literatures to invoke the research skills to design, interpret and make inferences from the resulting data.

**PO5**. Integrate and apply efficiently the contemporary IT tools to all computer applications.

**PO6**. Solve and work with a professional context pertaining to ethics, social, cultural and cyber regulations.

**PO7**. Involve in perennial learning for a continued career development and progress as a computer professional.

**PO8**. Function effectively both as a team leader and team member on multi-disciplinary projects to demonstrate computing and management skills.

**PO9**. Communicate effectively and present technical information in oral and written reports.

**PO10**. Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.

**PO11**. Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

**PO12**. Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

## PROGRAMME SPECIFIC OUTCOMES

**PSO1**. Design, develop and implement interdisciplinary application software projects to meet the demands of industry requirements using modern tools and technologies.

**PSO2**. To prepare graduates who will perform both as an individual and in a team through good analytical, design and implementation skills.

**PSO3**. To prepare graduates who will be lifelong learners through continuous professional development.

## FIRST SEMESTER

| Paper Code | Paper Title | L | P | Credit |
|---|---|---|---|---|
| **THEORY** | | | | |
| MCA-101 | Fundamentals of IT | 4 | 0 | 4 |
| MCA-103 | Problem solving using C Programming | 4 | 0 | 4 |
| MCA-105 | Discrete Mathematics | 4 | 0 | 4 |
| MCA-107 | Computer Organization | 4 | 0 | 4 |
| MCA-109 | Soft Skills | 4 | 0 | 4 |
| **PRACTICALS** | | | | |
| MCA-151 | Fundamentals of IT | 0 | 2 | 1 |
| MCA-153 | Problem solving using C Programming Lab | 0 | 4 | 2 |
| MCA-155 | Computer Organization Lab | 0 | 2 | 1 |
| MCA-157 | Linux Programming Lab | 0 | 2 | 1 |
| | **TOTAL** | 20 | 10 | 25 |

## SECOND SEMESTER

| Paper Code | Paper Title | L | P | Credit |
|---|---|---|---|---|
| **THEORY** | | | | |
| MCA-102 | Data and File Structures | 4 | 0 | 4 |
| MCA-104 | Object Oriented Programming in C++ | 4 | 0 | 4 |
| MCA-106 | Operating Systems | 4 | 0 | 4 |
| MCA-108 | Web Technology | 4 | 0 | 4 |
| MCA-110 | System Analysis and Design | 4 | 0 | 4 |
| **PRACTICALS** | | | | |
| MCA-152 | Data and File Structure Lab | 0 | 2 | 1 |
| MCA-154 | Object Oriented Programming in C++ Lab | 0 | 2 | 1 |
| MCA-156 | Web Technology Lab | 0 | 2 | 1 |
| MCA-158 | System Analysis and Design Lab | 0 | 2 | 1 |
| MCA-162 | Technical Report Writing* | 0 | 2 | 1 |
| | **TOTAL** | 20 | 10 | 25 |

## THIRD SEMESTER

| Paper Code | Paper Title | L | P | Credit |
|---|---|---|---|---|
| **THEORY** | | | | |
| MCA-201 | Software Engineering | 4 | 0 | 4 |
| MCA-203 | Database Management System | 4 | 0 | 4 |
| MCA-205 | Java Programming | 4 | 0 | 4 |
| MCA-207 | Data Communication and Networking | 4 | 0 | 4 |
| MCA-209 | Design and Analysis of Algorithms | 4 | 0 | 4 |
| **PRACTICALS** | | | | |
| MCA-251 | Software Engineering Lab | 0 | 2 | 1 |
| MCA-253 | Database Management System Lab | 0 | 2 | 1 |
| MCA-255 | Java Programming Lab | 0 | 2 | 1 |
| MCA-257 | Design and Analysis of Algorithms Lab | 0 | 2 | 1 |
| MCA-261 | Human Values and Professional Ethics* | 0 | 2 | 1 |
| | **TOTAL** | 20 | 10 | 25 |

## FOURTH SEMESTER

| Paper Code | Paper Title | L | P | Credit |
|---|---|---|---|---|
| **THEORY** | | | | |
| MCA-202 | Computer Graphics and Multimedia Technologies | 4 | 0 | 4 |
| MCA-204 | Business Intelligence | 4 | 0 | 4 |
| MCA-206 | Theory of Computation | 4 | 0 | 4 |
| MCA-208 | Cloud Computing | 4 | 0 | 4 |
| MTIT-614 | Business Analytics and BIG Data | 4 | 0 | 4 |
| **PRACTICALS** | | | | |
| MCA-252 | Computer Graphics and Multimedia Technologies Lab | 0 | 4 | 2 |
| MCA-254 | Business Intelligence Lab | 0 | 2 | 1 |
| MCA-256 | Business Analytics and BIG Data | 0 | 2 | 1 |
| MCA-258 | Cloud Computing Lab | 0 | 2 | 1 |
| | **TOTAL** | 20 | 10 | 25 |

## FIFTH SEMESTER

| Paper Code | Paper Title | L | P | Credit |
|---|---|---|---|---|
| **THEORY** | | | | |
| MCA-301 | Advanced Database Management Systems | 4 | 0 | 4 |
| MCA-303 | Software Testing and Quality Assurance | 4 | 0 | 4 |
| MCA-305 | Network Security | 4 | 0 | 4 |
| **Elective - I (Choose any One)** | | | | |
| MCA-307 | Numerical and Scientific Computing | 4 | 0 | 4 |
| MCA-309 | Mobile Computing | | | |
| MCA-311 | Artificial Intelligence | | | |
| MCA-313 | Microprocessors | | | |
| MCA-315 | Compiler Design | | | |
| MTIT-713 | E-Commerce and M-Commerce | | | |
| MCA-317 | Software Project Management | | | |
| **Elective - II (Choose any One)** | | | | |
| MCA-319 | Distributed Systems and Parallel Processing | 4 | 0 | 4 |
| MCA-321 | Organizational Behavior | | | |
| MCA-323 | Advanced Computer Architecture | | | |
| MCA-325 | Digital Signal Processing | | | |
| MCA-327 | Soft Computing | | | |
| MTCS-601 | Mobile Architecture and Programming | | | |
| MCA-329 | Emerging Trends | | | |
| **PRACTICALS** | | | | |
| MCA-351 | Advanced Database Management Systems Lab | 0 | 2 | 1 |
| MCA-353 | Software Testing & Quality Management Lab | 0 | 2 | 1 |
| MCA-355 | Network Security Lab | 0 | 2 | 1 |
| MCA-357 | Lab based on Elective – I & II | 0 | 2 | 1 |
| MCA-361 | Term Paper* | - | 2 | 1 |
| | **TOTAL** | 20 | 10 | 25 |

## SIXTH SEMESTER

| Paper Code | Paper Title | L | P | Credit |
|---|---|---|---|---|
| MCA-302 | Dissertation | - | - | 26 |
| MCA-362 | Seminar and Progress Report* | - | - | 4 |
| | TOTAL | - | - | 30 |

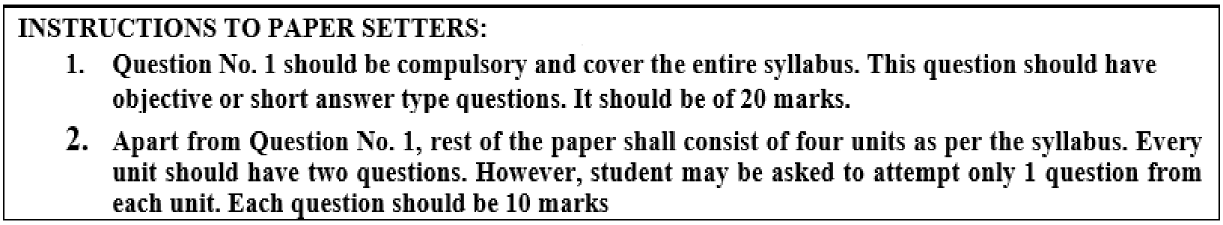
Non-University Examination System (NUES)

The total number of credits of the MCA Programme. = 155.

Each student shall be required to appear for examination in all courses. However, for the award of the degree a student shall be required to earn a minimum of 150.

# Fundamentals of Information Technology

| | |
|---|---|
| **Course Code:** MCA-101<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester**: 1 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

The course Fundamentals of Information Technology has become essential in the present age of computer technology and information, as the applications of information technology can be found in all aspects of our lives. This course is designed to meet the requirements of students having very little knowledge of computers and help them to learn from the basic fundaments of computers through applications of information technology.

## Course Objectives:

- To introduce skills relating to IT basics, computer applications, programming, Operating systems and computer network basics etc.
- To help students to understand specialized advanced courses in the Information Technology.

**Pre-requisite:** Preliminary knowledge of computer, their operations and applications.

## Course Outcomes:

CO1: Understanding the concept of input and output devices of computers.

CO2: Learn the functional unit and classify the type of computers, how they process information and how the individual computer interact with the other computing system and devices.

CO3: Understand an operating system and its working, and solve the common problems related to operating system.

CO4: Study to use the computer safely, legally, and responsibly.

## Pedagogy:

Lectures will be delivered via discussions, whiteboard, slideshows and assignments.

| UNIT-I | 10 hrs |
|---|---|

**Information Concepts and Processing:** Definition of Information Technology, Quality, need of information system, levels of information, data processing, definition of knowledge, Range of application: Scientific, business, educational, e-commerce, web publishing, Management Information System, Decision Support System, inventory control, and industrial control.

**Number System**: Bit, byte, binary, decimal, hexadecimal, and octal systems, conversion from one system to the other, Binary Arithmetic: Addition, subtraction and multiplication.

**Representation of Information:** Integer and floating-point representation, Complement schemes, Character codes (ASCII, EBCDIC, BCD, Excess-3, Grey).

| UNIT-II | 10 hrs |
|---|---|

**Introduction to Computer software:** Introduction to system software, categories of system and application, Distinction between systems software and Application software, Introduction to Software Development activities (Requirement, Design (algorithm and flowchart), Coding, Testing, Installation & Maintenance).

**Introduction to Computer Hardware**: CPU, Memory, different types of memories (Cache memory, virtual memory and Auxiliary memory) , Various I/O devices.

**Programming languages and Translators:** Low- and high-level languages, assembly language, 4GL and 5GL Introduction to assemblers, compilers, interpreters, linkers and loaders.

| UNIT-III | 10 hrs |
|---|---|

**Operating systems (Only introductory level):** Evolution, introduction to OS , functions and facilities, Different types of operating systems (Batch, multi-programming, time sharing, multiprocessing, PC operating system, real time operating system, single tasking and multitasking OS , single user and multi-user OS), Introduction to process management: process, threads, scheduling, characteristics of MS-DOS and Unix operating systems , DOS and UNIX commands, Introduction to Database Management System and its types.

| UNIT-IV | 10 hrs |
|---|---|

**Communication and Computer Network**: - Basic elements of a Communication System, , Data transmission media, Digital and Analog, Network Types (LAN, WAN and MAN), inter networking devices and Communication Protocols, Intranet and Extranet, Hypertext Markup Language, WWW, HTTP, HTTPs, FTP, Telnet, Web Browsers, Search Engines, Email, Digital Signatures, Firewall.

**Text Books**

1. Alex Leon and Mathews Leon, “Fundamentals of Information Technology”, Leon Techworld, 2007.
2. Robert G. Murdick, Joel E. Ross, “Introduction to management information systems”, Prentice Hall PTR.
3. A. S. Tananbaum, “Computer Networks”, 3rd Ed, PHI.
4. P. K. Sinha and Priti Sinha, “Computer Fundamentals”, BPB Publications, 2007.
5. Malvino and Leach, “Digital Principles and Application”, TMH,
6. D.H. Sanders,” Computers Today”, Mc Graw Hill.

**Reference Books**

1. Alex Leon and Mathews Leon, “Introduction to Computers”, Vikas Publishing House, 2007.
2. Norton Peter, “Introduction to computers”, TMH, 4th Ed., 2006.
3. Morris Mano, “Digital Design”, PHI, 2nd Ed, 2002.
4. Simon Haykins, “Communication System”, John Wiley & Sons, 2006.
5. B. Basaraj, “Digital Fundamentals”, Vikas Publications,
6. V. Rajaraman, “Introduction to Information Technology”, PHI, 2006.
7. V. Rajaraman, “Fundamentals of Computers”, PHI, 5th Ed., 2006.
8. David Anfinson and Ken Quamme, “IT Essentials PC Hardware and Software Component on Guide”, Pearson, 3rd Ed., 2008.

| Problem Solving using C Programming | |
|---|---|
| **Course Code:** MCA-103<br>**Contact Hours**: L-4 T-0 P-4<br>**Course Category:** DCC | **Credits:** 4<br>**Semester:** 1 |

## Introduction:

This course provides an introduction to computer concepts, logic, and computer programming. It includes designing, coding, debugging, testing, and documenting programs using a high-level programming language.

## Course Objectives

- To learn the fundamental programming concepts and methodologies, essential to build efficient C programs.
- To practice the fundamental programming methodologies in the C programming language via lab sessions.
- To code, document, test, and implement a well-structured, robust computer program using the C programming language.
- To write reusable modules (collections of functions) in C.

**Pre-requisite:** None

## Course Outcomes:

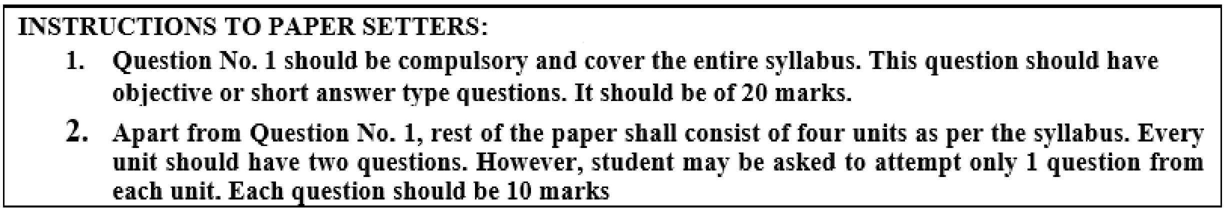
CO1: Recall the basic principles of C Programming.
CO2: Illustrate the use of Conditional Statements & Looping Concepts.
CO3: Develop the Concepts of programming Language.
CO4: Create a program using File operations.

**Pedagogy:** The class will be taught using theory and tutorial-based methods which include board teaching and presentations/slides, discussions etc. Along with classroom teaching, students will also be given assignments regarding the topics covered.

| **UNIT I** | **11 hrs** |
|---|---|
| **Introduction to Programming and its Environment:** Need for programming, Levels (High and Low) of programming, Development process (Preprocessor, Compiler, Linker and Loader), Linux –commonly used commands like mkdir, cd, ls, etc. , compiler –gcc, editor –vim<br>**C Language Introduction:** Program Structure through simple C programs, Constants and Variables, Data Types –Basic and Advanced, Operators and Expressions, Managing input and output operations using printf and scanf, Command line input, Conditional constructs, Looping constructs. Problem solving exercises based on –conditional and looping constructs | |
| **UNIT II** | **11 hrs** |
| **Pointers, Arrays and Strings:** Concept of memory, Definition, Usage –address of and value at operation, Pointer arithmetic. Pointer to pointer, Arrays (Single and Multi-dimensional) and Strings–with emphasis on role of pointers in them, Pointer to Array, Array of pointers. Problem solving exercises based on –pointers, arrays and strings.<br>**Procedural programming:** Functions (Function Prototyping, passing parameters through call by value and call by reference, returning values, recursion), Program organization using functions, Emphasis on reusability through C examples. Problem solving exercises based on –functions. | |
| **UNIT III** | **10 hrs** |
| **File handling:** Concept of streams, File pointer, Reading and Writing to file, Closing a file, Random access in a file, Error handling during file I/O operations. Problem solving exercises based on –files.<br>**Problem Solving:** Algorithm, Flowchart and Pseudo code. Program design. | |
| **UNIT IV** | **10 hrs** |
| **Advanced concepts:** Pointers to functions and Callback functions. Storage classes (auto, extern, static, register), The C Preprocessor (#define, #undef, #include, #if conditional inclusion and other preprocessor directives), Defining New Data Types–Structures, Unions, Enumerated Types<br>**Dynamic Memory Management:** malloc, calloc, realloc, size of, free.<br>**Introduction to Data Structure**: Linked Lists and dynamic data structures. Problem solving exercises based on –advanced concepts and data structure | |
| **Text Books** | |
| 1. Yashwant Kanetkar, “Let us C”, BPB Publications, 16th edition, 2018. | |
| 2. B. Kernighan and D. Ritchie, “The ANSI C Programming Language”, 2nd edition. | |
| **Reference Books** | |
| 1. Paul Deitel and Harvey Dietel, “How to Program”, PHI, 8th Ed., 2015. | |
| 2. Behrouz A. Forouzan and Richard F. Gilberg, “Computer Science A Structured Programming Approach Using C”, PHI, 3rd Ed., 2007 | |
| 3. Jeri R. Hanley and Elliot B. Koffman, “Problem Solving and Programming in C”, Pearson, 8th Ed. 2015. | |

# Discrete Mathematics

| | |
|---|---|
| **Course Code:** MCA-105<br>**Contact Hours**: L-4 P-4 | **Credits:** 4<br>**Semester:** 1 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

Discrete mathematics forms the mathematical foundation of computer and information science. This course familiarizes with a broad range of mathematical objects like sets, functions, relations, graphs, that are omnipresent in computer science.

## Course Objectives:

- To explain formal statements and their proofs; coming up with rigorous proofs themselves; and coming up with interesting results.
- To show at least one interesting and non-trivial result and give a full proof of introduced concepts.

**Pre-requisites:** Basic mathematical operations

**Course Outcome:** After studying this course, students will be able to:

CO1: Recall the basic principles of C Programming.

CO2: Illustrate the use of Conditional Statements & Looping Concepts.

CO3: Develop the Concepts of programming Language.

CO4: Create a program using File operations.

## Pedagogy:

The materials are delivered mostly through lectures videos to make complex subject easy to comprehend. More details on certain lessons are delivered through examples to provide more explanation.

| UNIT I | 10 hrs |
|---|---|
| **Introduction to Programming and its Environment:** Need for programming, Levels (High and Low) of programming, Development process (Pre-processor, Compiler, Linker and Loader), Linux – operating system familiarity, commonly used commands like mkdir, cd, ls, etc., compiler – gcc, editor – vim, use of debugger – gdb (to be taught throughout the course for debugging C programs)<br>**C Language Introduction:** Program Structure through simple C programs, Constants and Variables, Data Types – Basic and Advanced, Operators and Expressions, Managing input and output operations using printf and scanf, Command line input, Conditional constructs, Looping constructs. Problem solving exercises based on – conditional and looping constructs. | |
| **UNIT II** | **10 hrs** |
| **Pointers, Arrays and Strings:** Concept of memory, Definition, Usage – address of and value at operation, Pointer arithmetic. Pointer to pointer, Arrays (Single and Multi-dimensional) and Strings with emphasis on role of pointers in them, Pointer to Array, Array of pointers. Problem solving exercises based on – pointers, arrays and strings.<br>**Procedural programming:** Functions (Function Prototyping, passing parameters through call by value and call by reference, returning values, recursion), Program organization using functions, Emphasis on reusability through C examples. Problem solving exercises based on – functions. | |
| **UNIT III** | **10 hrs** |
| **File handling:** Concept of streams, File pointer, Reading and Writing to file, Closing a file, Random access in a file, Error handling during file I/O operations. Problem solving exercises based on – files.<br>**Problem Solving:** Algorithm, Flowchart and Pseudocode. Program design. | |
| **UNIT IV** | **11 hrs** |
| **Advanced concepts:** Pointers to functions and Callback functions. Storage classes (auto, extern, static, register), The C Preprocessor (#define, #undef, #include, #if conditional inclusion and other preprocessor directives), Defining New Data Types – Structures, Unions, Enumerated Types.<br>**Dynamic Memory Management:** malloc, calloc, realloc, size of, free. Introduction to Data Structure-Linked Lists and dynamic 2- dimensional arrays. Problem solving exercises based on – advanced concepts and data structures | |
| **Text Books** | |
| 3. E. Balaguruswamy, "Programming in ANSI C", 8th Edition, TMH. | |
| 4. Yashwant Kanetkar, "Let us C", BPB Publications, 2021. | |
| 5. B. Kernighan and D. Ritchie, "The ANSI C Programming Language", PHI. 2000. | |
| **Reference Books** | |
| 4. Yashwant Kanetkar, "Pointers in C", BPB Publications, 2002. | |
| 5. Paul Deitel and Harvey Dietel, "How to Program", PHI, 6th Ed., 2010. | |
| 6. R.P. Grimaldi, Discrete and Combinatorial Mathematics, 5/e, Addison Wesley, 2004. | |
| 7. Programming Approach Using C", PHI, 3rd Ed., 2007. | |

# Computer Organization

| | |
|---|---|
| **Course Code:** MCA-107<br>**Contact Hours:** L-4 P-0 | **Credits**: 4<br>**Semester:** 1 |

**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

The course aims to provide students with an understanding of the design of fundamental blocks of a computer system and interfacing techniques of these blocks to achieve different configurations of a computer system. It covers the basic topics in the design of computational units, instruction organization, memory systems, control and data flow, and interconnections.

## Course Objective:

- To have a thorough understanding of the basic structure and operation of a digital computer.
- To study the different ways of communicating with I/O devices and standard I/O interfaces.

**Pre-requisite:** Digital Systems and Computer Design

## Course Outcome:

CO1: Understand different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.

CO2: Comprehend the theory and architecture of central processing unit, pipelining, interrupt handling and memory organization

CO3: Analyze some of the design issues in terms of speed, technology, cost, performance.

CO4: Design combinational circuits for basic components of computer system and applications using multiplexers, decoders, flip flops etc.

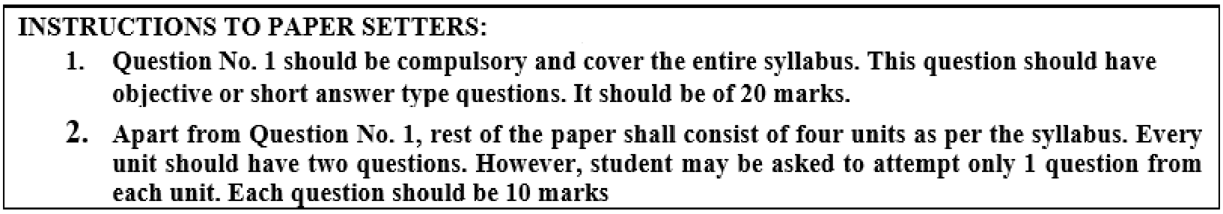
## Pedagogy:

The class will be taught using theory and tutorial-based methods which includes board teaching and presentations/slides, case studies, discussions etc. Along with classroom teaching, students will also be given assignments regarding the topics covered. The course instructor will demonstrate and explain about applications of Computer organisation techniques with research orientation.

| **UNIT-I** | **12 hrs** |
|---|---|
| **Digital Logic Circuit:** Basic Logic functions, Synthesis of logic functions using basic and universal gates, Boolean Algebra Properties, Flip-Flops, Registers, Shift- Registers, Counters, Decoders, Multiplexers, Functional Unit of computer system. **Data Representation:** Data types, R & (R-1)'s Complements, Fixed-Point representation, Floating point representation.<br>**Register Transfer and Micro operations:** Register transfer language, register transfer, Bus and Memory transfer, Arithmetic Micro operations, Logic Micro operations, Shift Micro operations | |
| **UNIT-II** | **10 hrs** |
| **Basic Computer Organization and Design:** Instruction Codes, Computer Instructions, Timing and Control, Instruction Cycle, Memory Reference Instructions, Input-Output and Interrupt.<br>**Micro programmed Control:** Control Memory. **Central Processing Unit:** Stack Organization, Instruction Formats, Addressing Modes, Program Control, Reduced Instruction Set Computer: RISC characteristics, CISC characteristics. Performance and Metrics. | |
| **UNIT-III** | **10 hrs** |
| **Pipelining and Vector Processing:** Parallel Processing, Pipelining, Arithmetic Pipelining, Instruction Pipelining, RISC Pipelining, Vector Processing, Array Processors. **Computer Arithmetic:** Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating- Point Arithmetic Operations. | |
| **UNIT-IV** | **10 hrs** |
| **Input-Output Organization:** Peripheral Devices, Input-Output interface, Asynchronous data transfer, Modes of transfer, Priority Interrupt, Direct Memory Access. **Memory organization:** Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware. | |
| **Text Books** | |
| 1. M. Morris Mano, "Computer System Architecture", PHI, 3rd Edition, 2016. | |
| 2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", McGraw Hill, 5 th Edition, 2012. | |
| 3. William Stallings, "Computer Organization and Architecture", PHI, 11th edition, 2021. | |
| **Reference Books** | |
| 1. John L. Hennessy and David A. Patterson, "Computer Architecture a quantitative approach", Elsevier, 6th Edition, 2019. | |
| 2. Anand kumar, "Fundamentals of digital circuits", PHI, 4th edition, 2016 | |

# Soft Skills

| | |
|---|---|
| **Course Code:** MCA-109<br>**Contact Hours:** L-4 P-0 | **Credits**: 4<br>**Semester:** 1 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

This course aims to enhance the students' professional communication skills by providing adequate exposure in verbal and nonverbal skills and related sub skills. The course is designed to provide awareness of appropriate communication strategies with social, organizational and cultural awareness. The course empowers students in day-to-day professional soft skills like listening skills, presentation skills, and group discussion etc.

## Course Objectives:

- To know the process of professional communication and its various components.
- To improve language skills i.e. Listening Skills, Speaking Skills, Reading Skills and Writing Skills (LSRW).
- To create literary sensibility and enhance comprehension skills.
- To develop confidence for communicating in English language.

**Pre-requisites:** None

## Course Outcome:

CO1: Apply their fundamentals of communication and learn how to function effectively in multi- disciplinary environments.

CO2: Explain the importance of reading, interpreting, and writing skills to function efficiently as an individual.

CO3: Develop speaking skills to communicate effectively and to manage projects as a member or leader in diverse situations.

CO4: Determine the importance of giving and receiving clear instructions and function effectively in multi-disciplinary environments.

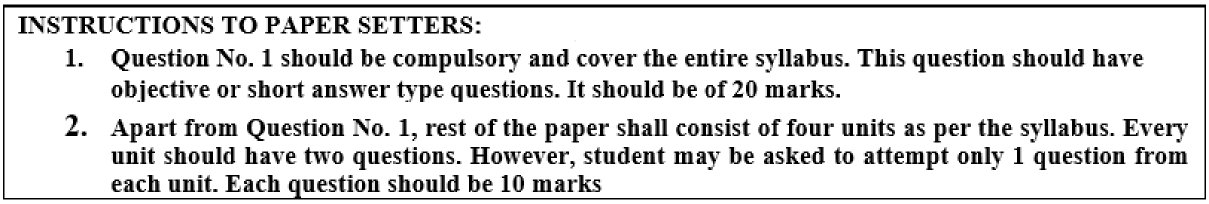
## Pedagogy:

To provide knowledge of various communication processes through innovative and interactive classroom teaching sessions. To evaluate students' progress through practical sessions including Group discussion, Presentations, role plays and JAMs.

| | |
|---|---|
| **UNIT-I** | **12 hrs** |
| **Types of Communication, Oral Communication:** clarity, speed, tone and pitch, Oral and Aural skills, Sounds, Introduction to syllable stress, Noun stress, Voiced and voiceless sounds, Diphthongs, Rate of speech, Vowel and consonant, Phonetics. Informal vs Formal communication, Communication Barriers.<br>**Language skills:** Vocabulary, Phrase, Clause, Sentence fluency building, Word match, reading aloud, Recognition of attributes, listening – reading comprehension, Listening Sills, Parts of speech, Media/channels for communication, Written Communication, Grammar. | |
| **UNIT-II** | **10 hrs** |
| **Self-analysis** through SWOT, Johari window, Personality Development, Intra personal communication vs. Inter personal Communication and Relationships, Leadership Skills, Team Building, Public speaking, Individual Communication, Self-advertising, Over stating and understating, Time Management.<br>**Communication Boosters:** Body language, Voice, Posture and gesture, Eye contact, Dress codes, Verbal crutches, Pronunciation, Contextualization: creating and understanding contexts, Aura words. | |
| **UNIT-III** | **10 hrs** |
| **Interview:** Types of Interviews, preparing for the Interviews, Attending the Interview, Interview Process, Employers Expectations, General Etiquette.<br>**Group Discussions:** Guidelines, Expressions, Evaluation. Video conferencing, Telephone skills,<br>**Teleconferencing, Participation in meetings:** chairing sessions. Presentation Skills, Types of presentation, Capturing Data, Guidelines to make an effective presentation, Body Language, Voice Modulation, Integrating voice & picture, Audience Awareness, Presentation Plan, Visual Aids, Forms of Layout, Styles of Presentation, Management presentations. | |
| **UNIT-IV** | **10 hrs** |
| **Letter writing:** Types of Letters, Business letters, E-mail, Fax, Pro-forma culture, Drafting the Applications, Format, Style, Effectiveness, study of sample letters, Elements of structure, Preparing a CV / Resume, Statement of Purpose, Web chat, Greeting, Memos, Reports, Minutes, Business correspondence. | |
| **Text Books** | |
| 1. Essentials of Business Communication, Rajendra Pal, JS Korlahhi. | |
| 2. Basic Communication Skills for Technology, Andre J. Rutherford: Pearson Education Asia. | |
| 3. KR Lakshiminarayana: English for Technical Communication. | |
| **Reference Books** | |
| 1. Business Communication, RK Madhukar, Vikas Publishing House Pvt. Ltd. | |
| 2. Edmund H Weiss: Writing Remedies: Practical Exercises for Technical Writing. Universities Press. | |

# Data and File Structures

| | |
|---|---|
| **Course Code:** MCA-102<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 2 |



INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks

## Introduction:

This course covers the design, analysis, and implementation of data structures and algorithms to solve engineering problems using an object-oriented programming language. Topics include elementary data structures, (including arrays, stacks, queues, and lists), advanced data structures (including trees and graphs), the algorithms used to manipulate these structures, and their application to solving practical engineering problems.

## Course Objectives:

- To learn efficient storage mechanisms of data for an easy access.
- To design and implement various basic and advanced data structures.
- To introduce various techniques for representation of the data in the real world.
- To develop applications using data structures.

**Pre-requisite:** Standard programming language C/C++, mathematical knowledge, knowledge of basic probability.

**Course Outcomes:** Upon successful completion of this course, students will be able to:

CO1: Understand the basics of data structures to represent data items in the real world.

CO2: Evaluate the time and space complexities of Algorithms.

CO3: Apply and implement the application of sorting and pattern-matching algorithms.

CO4: Create projects using a variety of data structures such as stacks, queues, hash tables, binary trees, search trees, heaps, graphs, and B-trees.

**Pedagogy:** The class will be taught using theory and tutorial-based methods which include board teaching, presentations/slides, discussions and case-based studies. Along with classroom teaching, students will also be given assignments regarding the topics covered.

| **UNIT-I** | **11 hrs** |
|---|---|
| **Introduction:** Abstract Data Type, Elementary Data Organization, Measuring efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations. Arrays: Single and Multidimensional Arrays,<br>**Representation of Arrays:** Row Major Order, Column Major Order, Application of arrays, Sparse Matrices.<br>**Linked lists:** Array and Dynamic Implementation of Single Linked Lists, Doubly Linked Lists, Circularly Linked Lists, and Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation, and Addition.<br>**Stacks:** Stack operations: Push & Pop, Array and Linked list implementation of Stack, Applications: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion. | |
| **UNIT-II** | **11 hrs** |
| **Queues:** Operations: Create, Add, Delete, full and empty queues, Array and linked implementation of queues, Dequeue, Circular queues, and Priority Queue. Hashing: Hash Function, Hash Table, Collision Resolution Strategies.<br>**Trees:** Basic terminology, Binary Trees, Array, and linked list implementation, Types of Binary Tree, Extended Binary Trees, Algebraic Expressions, Tree Traversal algorithms: Inorder, Preorder, and Postorder, Threaded Binary trees, Search, Addition and deletion of an element in a binary tree, AVL Trees, Heaps, B Trees, Trees, and their applications, Evaluating an expression tree. | |
| **UNIT-III** | **10 hrs** |
| **Searching:** Sequential search, Binary Search. Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Merge Sort, Heap Sort, Radix Sort, Bucket Sort, Shell Sort, Graphs: Representation (Matrix and Linked), Traversals, Shortest path, Topological sort. Dijkstra’s Algorithm, Floyd Warshall’s Algorithm, and Minimum Spanning Tree Algorithms (Kruskal’s Algorithm, Prim’s Algorithm). | |
| **UNIT-IV** | **10 hrs** |
| **Files:** Creation and Processing of files, File handling using command line arguments, File opening, closing, modes, formatted inputs, output to file, reading/writing of files, accessing records randomly, updating files. Operations on files, Library functions, File Indexing (primary, secondary, clustered, unclustered, dense, sparse), File streams, Hierarchy of file stream classes, and Error handling during file operations. | |
| **Text Books** | |
| 1. Aaron Tanenbaum, “Data Structures Using C”, Pearson Education India, 2nd edition, 2016/Latest Edition. | |
| 2. Ellis Horowitzand Sartaj Sahni, “Fundamentalsof data structures”, Silicon Pr, 2nd Edition, 2017/Latest Edition. | |
| **Reference Books/Materials** | |
| 1. Seymour Lipschutz, “Data Structures,” Cengage Learning, 2nd Edition, 2015/Latest Edition. | |
| 2. Donald Knuth, “The Art of Computer Programming,” Addison-Wesley,3rd Edition, 2015/Latest Edition. | |
| 3. https://nptel.ac.in/courses/106/106/106106145/ | |

# Object Oriented Programming in C++

| | |
|---|---|
| **Course Code:** MCA-104<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 2 |

**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

This course provides in-depth coverage of object-oriented programming principles and techniques. Topics include classes, objects, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes etc.The course material embraces the C++11 language standard with numerous examples demonstrating the benefits of C++11. In the end some basics of Java will be covered.

## Course Objective:

To learn object-oriented programming (OOP) principles and get a flavour of modular programming.

**Pre-requisite:** Basics of C Programming

## Course Outcomes:

After completion of the course, the students will be able:

CO1: Distinguish between the various programming paradigms available and understand the basic syntax of object-oriented programming.

CO2: Build the classes and apply the various features of the language.

CO3: Able to develop programs with reusability.

CO4: Implement program using namespace, templates, exception handling and file I/O to improve effective programming skills.

**Pedagogy:** Emphasis on lab sessions where students will be given programming assignments to code in C++/Python/Java based on topics learnt in previous lectures.

| **UNIT-I** | **10 hrs** |
|---|---|
| **Introduction to Object Oriented Programming**: Need for Object Oriented Programming, Comparison of Programming paradigms, Characteristics of Object-Oriented Programming Languages, Structure of a C++ program, Use of *cin* and *cout*, Compilation process.<br>**C++ Programming Language (Procedural):** Tokens, Data Types (Basic, Advanced and Derived), Variables, Reference vs Pointers, Operators (scope resolution, dynamic memory related, type cast), Expressions, Functions (inline function, *const* arguments, default arguments). | |
| **UNIT-II** | **10 hrs** |
| **Classes and Objects:** Objects, Classes, Encapsulation, Data Abstraction, Role of *private* and *public* access specifier, Memory organization of class, Member functions – inline and non-inline, static member variables, Friend functions, Class vs Structure, Constructors – default, parameterized, copy and dynamic, Destructors, Assignment operator – deep and shallow copying.<br>**Polymorphism:** Function overloading, Constructor overloading, Compile time polymorphism, Overloading Rules, Operator Overloading (Unary and Binary) as member function/friend function, Example operators to be overloaded: Arithmetic, Output/Input, Prefix/ Postfix Increment and Decrement, Comparison, Assignment, subscript and function call Operator. | |
| **UNIT-III** | **10 hrs** |
| **Inheritance:** Inheritance, Types of Inheritance, Use of protected access specifier, Virtual base class, Ambiguity resolution using scope resolution operator and Virtual base class, Overriding inheritance methods, Constructors and Destructor in derived classes. Runtime polymorphism, Pointer to objects, Virtual Functions (concept of virtual table), pure virtual functions, Abstract Class.<br>**Managing Input / Output:** Concept of streams, console I/O – formatted and unformatted, Manipulators, File I/O – Predefined classes, file opening & closing, file manipulation, read & write operations, sequential and random file access. | |
| **UNIT-IV** | **10 hrs** |
| **Exception Handling:** Basic mechanism, Throwing, Catching and Re-throwing.<br>**Namespace:** Basic concept, role of scope resolution operator and *using* keyword.<br>**Generic Programming:** User defined Templates - Class templates with and without multiple parameters and Function templates with and without parameters, Template overloading. Standard Template Library (STL): Introduction, Components – Container, Iterator and Algorithm, Example programs using STL. | |
| **Text Books** | |
| 1. E. Balaguruswamy, “Object Oriented Programming with C++”, 4th Edition, TMH. | |
| 2. Bjarne Stroustrup , “The C++ Programming Language”, Addison Welsley, 3rd Ed. | |
| **Reference Books/Materials** | |
| 1. D . Parasons, “Object Oriented Programming with C++”, BPB Publication. | |
| 2. Steven C. Lawlor, “The Art of Programming Computer Science with C++”, Vikas Publication. | |
| 3. Schildt Herbert, “C++: The Complete Reference”, Tata McGraw Hill, 4th Ed., 1999. | |
| 4. Behrouz A. Forouan, Richrad F. Gilberg, Computer Science - A Structural Approach Using C++”, Cengage Learning, 2004. | |
| 5. Nell Dale, “C++ Plus Data Structure”, Jones and Bartlett, 4th Ed., 2010. | |
| 6. Nell Dale, Chips Weens, “Programming and Problem Solving with C++”, Jones and Bartlett , 5th Ed., 2010. | |

# Operating Systems

| | |
|---|---|
| **Course Code:** MCA-106<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 2 |

**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

This course aims at introducing classical internal algorithms and structures of modern operating systems including CPU scheduling, memory management, and device management. Topics including file systems, virtual memory, disk scheduling, concurrent processe s, deadlocks, security, and integrity will be covered.

## Course Objectives:

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication.
- To learn the mechanisms involved in memory management in contemporary OS.
- To gain knowledge on OS architecture, mutual exclusion algorithms, deadlock detection algorithms etc.
- To know the components and management aspects of concurrency management.
-

**Pre-requisite**: Analysis of algorithms, algorithm design techniques, programming knowledge in C, C++ or JAVA.

**Course Outcome**: After studying this course, students will be able:

CO1: To understand various types of OS, basic concepts, various functions of different OS, process management & CPU scheduling.

CO2: To compare and contrast various memory management schemes like paging, segmentation and to apply different deadlock handling algorithms.

CO3: To implement different disk scheduling algorithms, to apply and use various process synchronization techniques and device management strategies.

CO4: To understand management of I/O and different file handling & directory implementation schemes in OS.

**Pedagogy:** The class will be taught using theory and tutorial-based methods which includes board teaching and presentations/slides, discussions, case studies etc. Along with classroom teaching, students will also be given assignments regarding the topics covered. The course instructor will demonstrate and explain about applications of Operating Systems techniques with real-time examples.

| | |
|---|---|
| **UNIT-I** | **11 hrs** |
| **Introduction:** Introduction to Operating System, **Types of O.S:** Simple Batch, Multi-programmed Batched, Time-Sharing, Personal-computer, Parallel, Distributed, Real-Time, Mobile<br>**Operating-System Structures:** Layered Architecture, System Calls, System Programs, System Structure, Virtual Machine<br>**Processes:** Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Inter-process Communication, Threads, Multithreaded Programming.<br>**CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real-Time Scheduling | |
| **UNIT-II** | **11 hrs** |
| **Process Synchronization:** Background, Critical-Section Problem, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Critical Regions, Monitors.<br>**Memory Management:** Background, Logical versus Physical Address space, Swapping, Contiguous allocation, Fragmentation, Paging, Segmentation, Segmentation with Paging.<br>**Virtual Memory:** Demand Paging, Page Replacement, Page-replacement Algorithms, Performance of Demand Paging, Allocation of Frames, thrashing.<br>**Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock | |
| **UNIT-III** | **10 hrs** |
| **Device Management:** Techniques for Device Management, Dedicated Devices, Shared Devices, Virtual Devices<br>**Secondary-Storage Structure:** Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, Disk Reliability, Stable-Storage Implementation | |
| **UNIT-IV** | **10 hrs** |
| **Information Management:** Introduction, Simple File System, General Model of a File System, Symbolic File System, Basic File System, Access Control Verification, Logical File System, Physical File System<br>**File-System Interface:** File Concept, Access Methods, Directory Structure, Protection, and Consistency Semantics. **File-System Implementation**: File-System Structure, Allocation Methods, Free-Space Management, Directory Implementation, Efficiency and Performance, Recovery. | |
| Text Books | |
| 1. Silberschatz and Galvin, “Operating System Concepts”, John Wiley, 9th Ed., 2016. | |
| 2. Madnick E. and Donovan J., “Operating Systems”, Tata McGraw Hill, 2017. | |
| 3. Tannenbaum, “Operating Systems”, PHI, 5th Ed. | |

# Web Technology

| | |
|---|---|
| **Course Code:** MCA-108<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 2 |

**Introduction:** This course aims at introducing the fundamental of internet and concepts of web technology.

## Course Objectives:

- To understand the basics of Internet and the Web phenomena.
- To create the web pages and essential areas of developing the website.
- To introduce PHP language for server-side scripting
- To introduce XML and processing of XML Data
- To introduce Client-side scripting with Javascript and AJAX

**Pre-requisites:** Basic knowledge of programming.

**Course Outcomes:** Upon successful completion of this course, students will be able to:

CO1: Design and develop web applications.

CO2: Design and Explain the basic concept of XML and Create XML documents and Schema.

CO3: Develop web-based application using suitable client side and server-side web technologies

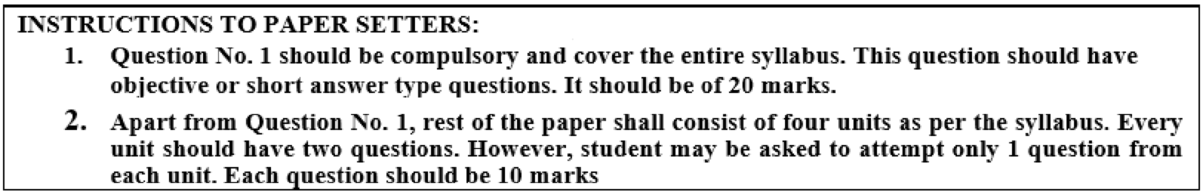
CO4: Develop solution to complex problems using appropriate method, technologies, frameworks and web services.

**Pedagogy:** Students will design web pages using static and dynamic pages, with the introduction on client side and server-side programming. Emphasis on developing web applications. The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web-based resources as well as flipped class room teaching will be adopted.

| |
|---|
| **UNIT 1**<br>**10hrs** |
| **Introduction to the internet, the world wide web:** the idea of hypertext and hyper media, how the web works, how the browser works- mime types, plugins and helper applications. **Introduction to html**: basic tags of html, tables, frames, forms. **Separating style from structure with style sheets:** inline style specification and internal style specifications within html, external linked style specification using css. |
| **UNIT 2** **10hrs** |
| **Introduction to xml**: XML vs. HTML, uses of xml, simple xml, xml key components, dtd and schemas, well formed, using xml with application. **Client-side programming:** introduction to JavaScript, JavaScript programming, variables, functions, conditions, loops, JavaScript object model, event handling, forms handling, cookies, hidden fields, images, applications. |
| **UNIT 3** **10hrs** |
| **DHTML**: combining html, css and javascript, dhtml document object model (dom). **SQL Queries:** Introduction to SQL**,** Simple queries with use of where, having, group by, View, create, drop. **Server-side programming:** Introduction to php, basics of php, php file handling, php file upload, php sessions, php cookies, php error handling, php mysql introduction, php mysql insert into, php mysql select, php mysql -the where clause, php mysql update, php mysql delete. |
| **UNIT 4** **10hrs** |
| **Web services:** components and working of web services, web services architecture, introduction to service-oriented architecture, soap, wsdl, uddi, ajax, overview of grid computing, overview of cloud computing. |
| **Text Books:** |
| 1. Deitel, “Internet and World Wide Web, How to Program”, PHI |
| 2. Achyut S Godbole and Atul Kahate, “Web Technologies”, Tata McGraw Hill. |
| 3. Ivan Bay Ross, “HTML, DHTML, Java Script, Perl CGI”, PBP |
| **References:** |
| 1. Jeffrey C. Jackson, “Web Technologies – A Computer Science Perspective”, Pearson, |
| 2. Anders Moller, Michael Schwartzeach, “An Introduction to XML and Web Technologies”, Pearson, 2009 |
| 3. James L Mohler and Jon Duff, “Designing Interactive web sites”, Delmar Thomson Learning. |

# Systems Analysis & Design

| | |
|---|---|
| **Course Code:** MCA-110<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 2 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

Systems Analysis and Design provides students with concepts of the analysis and design processes and allows students to use industry standard methodology and framework to develop business information systems. The course combines terminology with conceptual and practical approaches to designing and implementing business systems. Analytical and problem-solving skills are developed through a modern integrated, structured approach. Predictive and adaptive approaches to systems development life cycle (SDLC) using an iterative approach are covered. The course contains the entire analysis and design process from conception through implementation, including training and support, system documentation and maintenance, and relevant project management techniques. Tools and techniques to optimize performance and secure the system are introduced. Tools that optimize performance and secure the system include SDLC, unified process (UP), extreme programming (XP), and scrum.

## Course Objectives:

- Illustrate the duties and activities of a systems analyst.
- Explain the purpose and various phases of the systems development life cycle (SDLC).
- Demonstrate an understanding of project management.
- Assess analysis and design tools and techniques.
- Evaluate case studies for real-life aspects of systems analysis and design.
- Use one of the popular systems development processes.
- Evaluate the important aspects of training and user support.

**Course Outcome:** At the end of the course, the students will be able to

CO1: Define and use common System Analysis and Design fundamental terminology.

CO2: Utilize current Analysis and Design tools to graphically characterize processes and flows in a business system.

CO3: Design and create effective Input/Output including Web pages/forms.

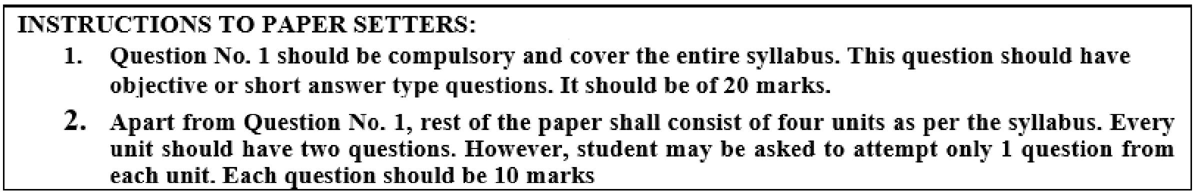
CO4: Design Logical Databases.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web- based resources as well as flipped classroom teaching will be adopted.

| **UNIT-I** | **10 hrs** |
|---|---|
| **Introduction :** System Definition and concepts, System Environments and Boundaries , Basic principles of successful systems, , Role and Need of Systems Analyst. Qualifications and responsibilities, System Analysis as a Profession.<br>**System Development Cycle**: Introduction to Systems Development Life Cycle (SDLC). Various phases of SDLC: Study, Analysis, Design, Development, Implementation, Maintenance, Systems documentation consideration, Enforcing documentation discipline in an organization. | |
| **UNIT –II** | **11 hrs** |
| **System Planning**: Data and fact gathering techniques: Interviews, Group Communication -Questionnaires; Assessing Project Feasibility: Technical, Operational, Economic, Cost Benefits, Analysis, Schedule, Modern Methods for determining system requirements: Joint Application, Development Program, Prototyping, Business Process Re-engineering. System Selection Plan and Proposal. | |
| **UNIT-III** | **11 hrs** |
| **System Design and Modelling**: Process Modelling, logical and physical design, Conceptual Data Modelling: Entity Relationship Analysis, Entity-Relationship Modelling, DFDs, Concepts of Normalization, Process Description: Structured English, Decision Tree, Decision Tables. Data Dictionary, Recording Data Descriptions, Module specifications. Top-down and bottom-up design. Module coupling and cohesion. Structure Charts. | |
| **UNIT –IV** | **10 hrs** |
| **Input and Output**: Classification of forms, Input/output forms design. User-interface design, Graphical interfaces. Standards and guidelines for GUI design. Designing integrity controls and security controls<br>**System Implementation and Maintenance**: Planning considerations. Conversion methods, procedures and controls. System acceptance criteria. System Evaluation and Performance. Testing and Validation. Preparing User Manual. Maintenance Activities and Issues. | |
| **Text Books** | |
| 1. J. Hoffer, "Modern Systems Analysis and Design", Fourth Edition, Joey George and Joseph Valacich, Pearson Education. | |
| 2. J. Whitten, L. Bentley and K. Dittman, "Systems Analysis and Design Methods", Fifth Edition, Tata McGraw Hill. | |
| **References Books** | |
| 1. Shelly, Cashman, Rosenblatt, "Systems Analysis and Design" Sixth Edition, Thompson Course Technologies. | |
| 2. Kendall & Kendall, "Systems Analysis and Design", Seventh Edition, Pearson | |

# Software Engineering

| | |
|---|---|
| **Course Code:** MCA-201<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 3 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks

## Introduction:

This course introduces students to the different software development lifecycle (SDLC) phases used in developing, delivering, and maintaining software products. Students will also acquire basic software development skills and understand common terminology used in the software engineering profession. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of software development projects.

## Course Objectives:

- To introduce the concepts of software engineering, software processes and its models.
- To understand the software requirements analysis, transform the requirements using DFD, create software requirement specification document and validation of the software requirements.
- To understand fundamental of software design, software quality and software maintenance.
- To understand the project planning process, size and cost estimation techniques for development of software.

**Pre-requisite:** Basic knowledge of Programming Languages.

**Course Outcomes**: Upon successful completion of this course, students will be able to

**CO1**: Understand the concepts of Software engineering, Software process and its models.

**CO2**: Evaluate the Software Requirements Specification, Interpret and Create Software Requirements Specification Document.

**CO3**: Apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices, evaluate the quality and maintenance of the software through software testing.

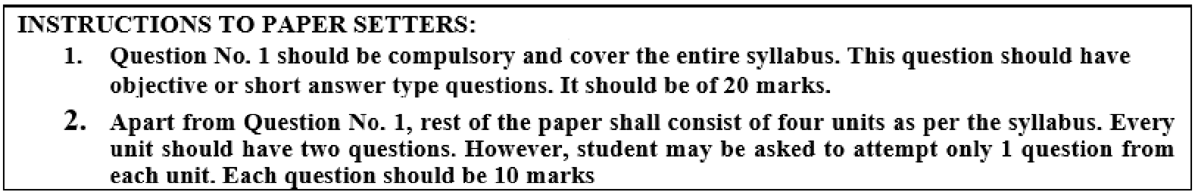
**CO4**: Create the software project plan for size and cost estimation including risk analysis.

**Pedagogy:** This course is structured around continuous progress. It will include a combination of lectures, and group activities focused on experiential learning, in-class discussions, regular assessments and case studies. The topics will be presented to students using real-world scenarios and problem-solving activities.

| | |
|---|---|
| **UNIT-I** | **10 hrs** |
| **Introduction:** Software Crisis, Software Processes & Characteristics, Software life cycle models, Waterfall, Prototype, Evolutionary and Spiral Models.<br>**Software Requirements analysis & specifications:** Requirement engineering, requirement elicitation techniques like FAST, QFD & Use case approach, requirements analysis using DFD, Data dictionaries & ER Diagrams, Requirements documentation, Nature of SRS, Characteristics & organization of SRS, Requirement Management, IEEE Std. for SRS. | |
| **UNIT –II** | **11 hrs** |
| **Software Project Planning:** Size Estimation like lines of Code & Function Count, Cost Estimation Models, COCOMO, Putnam resource allocation model, Validating Software Estimates, Risk Management.<br>**Software Design:** Cohesion & Coupling, Classification of Cohesiveness & Coupling, Function Oriented Design, Object Oriented Design. | |
| **UNIT-III** | **11 hrs** |
| **Software Metrics:** Software measurements: What & Why, Token Count, Halstead Software Science Measures, Data Structure Metrics, Information Flow Metrics.<br>**Software Reliability:** Importance, Hardware Reliability & Software Reliability, Failure and Faults, Reliability Models- Basic Model, Logarithmic Poisson Model, Software Quality Models, CMM & ISO 9001. | |
| **UNIT –IV** | **10 hrs** |
| **Software Testing:** Testing process, Design of test cases, Introduction to functional testing & Structural testing, Unit Testing, Integration and System Testing, Debugging, Alpha & Beta Testing.<br>**Software Maintenance:** Management of Maintenance, Maintenance Process, Maintenance Models, Regression Testing, Reverse Engineering, Software Re-engineering, Configuration Management, Documentation. | |
| **Text Books** | |
| 1. K. K. Aggarwal and Yogesh Singh, "Software Engineering", New Age International, 3rd Ed., 2005. | |
| 2. R. S. Pressman, "Software Engineering – A Practitioner's Approach", McGraw Hill Int. ,5th Ed., 2001. | |
| 3. Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa, 3rd Ed., 2005. | |
| **References Books** | |
| 1. Stephen R. Schach, "Classical & Object-Oriented Software Engineering", IRWIN,1996. | |
| 2. I. Sommerville, "Software Engineering", Addison Wesley,8th Ed., 2009. | |
| 3. Frank Tsui and Orlando Karan, "Essentials of Software Engineering", Joes and Bartlett,2nd Ed., 2010. | |
| 4. Kassem A. Saleh, "Software Engineering", Cengage Learning, 2009. | |
| 5. Rajib Mall, "Fundamrntal of Software Engineering", PHI, 3rd Ed., 2009. | |

# Database Management System

| | |
|---|---|
| **Course Code:** MCA-203<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 3 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** Database Management System (DBMS) is used for creating and managing the databases. The main aim of a DBMS is to supply a way to store-up and retrieve the desired database information as per the application requirement, which is both convenient and efficient.

## Course Objectives:

- Describe the fundamental elements of relational database management systems, relational data model, entity-relationship model, relational database design, relational algebra and SQL.
- To design relational databases by applying normalization techniques to normalize the database.
- Strong practice in SQL programming through a variety of database problems.
- Understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.

**Pre-requisites:** Basic concepts of set theory.

**Course Outcomes:** Upon completion of the course, the students will be able:

CO1: To have a high-level understanding of major DBMS components and their functions.

CO2: To model an application’s data requirements using conceptual modeling tools like ER diagrams and design database schemas based on the conceptual model.

CO3: To develop structured query language (SQL) queries to create, read, update, and delete relational database data.

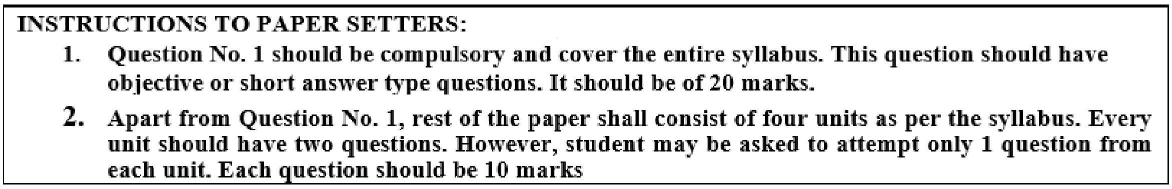
CO4: To understand the concept of Transaction, concurrency and Query processing.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web-based resources as well as flipped class room teaching will be adopted.

| **UNIT I** | **10hrs** |
|---|---|
| **Introduction:** Database system concepts and its architecture, Data models schema and instances, Data independence and database language and interface, Data definition languages, DML. Overall database structure.<br>**Data modeling using Entity Relationship Model:** ER model concept, notation for ER diagrams mapping constraints, Keys, Concept of super key, candidate key, primary key generalizations, Aggregation, reducing ER diagrams to tables, extended ER model.<br>**Relational Data Model and Language:** Relational data model concepts, integrity constraints, Keys domain constraints, referential integrity, assertions, triggers, foreign key. | |
| **UNIT II** | **12hrs** |
| Relational algebra, relational calculus, SQL Queries, SQL Functions, Nested Queries, Joins, Advanced Queries, Views, Indexing, Sequence, Grant and Revoke, Materialized View, Introduction to PL/SQL | |
| **UNIT III** | **10hrs** |
| **Data Base Design:** Functional dependencies, normal forms, 1NF, 2NF, 3NF and BCNF, multi-valued dependencies fourth normal form, join dependencies and fifth normal form. Inclusion dependencies, lossless join decompositions, normalization using FD, MVD and JDs, Denormalization. | |
| **UNIT IV** | **10hrs** |
| **Transaction processing concepts:** Transaction processing system, schedule and recoverability, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recovery from transaction failures, deadlock handling.<br>**Concurrency Control Techniques:** Locking Techniques for concurrency control, time stamping protocols for concurrency control, concurrency control in distributed systems. Multiple granularities and multi-version schemes. | |
| **Text Books** | |
| 1. Elmasri Ramez and Navathe Shamkant, Fundamentals of Database System, Pearson, 7th Ed. (June 2017) | |
| 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, McGraw Hill, 7th Ed(2019) | |
| **Reference Books** | |
| 1. Ceri and Pelagatti, Distributed Databases: Principles & Systems, McGraw-Hill, 2017. | |
| 2. Conolly & Begg, Database Management Systems, Pearson Education Asia., 5th Edition, 2010 | |

# Java Programming

| | |
|---|---|
| **Course Code:** MCA-205<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 3 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** Java Programming is one of the most widely used programming language among developers and are preferred over other languages. This course introduces students to object-oriented design methods and GUI like Applet, swing, AWT etc. The objective is to provide students with the use of the Java programming language for writing complex and stand-alone applications at the Intermediate level.

**Course Objectives:**

- To provide knowledge of Object-Oriented programming features and fundamentals of program development using java.
- Students will learn how to write, test, and debug Object-Oriented programs using Java and learn advanced concepts.

**Pre-requisites**: The student may have experience in a high-level programming language such as C/C++.

**Course Outcomes:** Upon completion of the course, the students will be able:

- Understand object-oriented concepts and use the concepts of inheritance, polymorphism, interfaces, and packages to create classes and reusable Java programs.
- Identify operations commonly used to implement thread-based applications, file I/O operations, and exception handling
- Implement simple GUI interfaces for a computer program to interact with users and understand the event handling.
- Understand the basic concepts of networking to develop network-based applications and learn JDBC.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web-based resources as well as flipped class room teaching will be adopted.

| | |
|---|---|
| **UNIT I** | **10hrs** |
| **Overview and characteristics of Java**: Java program Compilation and Execution Process Organization of the Java Virtual Machine, JVM as an interpreter and emulator, Instruction Set, class File Format, Verification, Class Area, Java Stack, Heap, Garbage Collection. Security Promises of the JVM, Security Architecture and Security Policy. Class loaders and security aspects, sandbox model<br>**Introducing classes, objects and methods:** defining a class, adding variables and methods, creating objects, constructors, class inheritance. Arrays and String: Creating an array, one- and two-dimensional arrays, string array and methods, Classes: String and String Buffer classes, Wrapper classes: Basics types, using super, Multilevel hierarchy abstract and final classes, Object class, Packages and interfaces, Access protection, Extending Interfaces, packages. | |
| **UNIT II** | **12hrs** |
| **Exception Handling:** Fundamentals exception types, uncaught exceptions, throw, throw, final, built in exception, creating your own exceptions, Multithreaded Programming: Fundamentals, **Java thread model:** priorities, synchronization, messaging, thread classes, Runnable interface, inter thread Communication, suspending, resuming and stopping threads. Input/Output Programming: Basics, Streams, Byte and Character Stream, predefined streams, Reading and writing from console and files.<br>**Using Standard Java Packages (lang, util, io, net). Networking:** Basics, networking classes and interfaces, using java.net package, doing TCP/IP and Data-gram Programming, RMI (Remote Method Invocation). | |
| **UNIT III** | **10hrs** |
| **Event Handling:** Different Mechanism, the Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and Inner Classes, Working with windows, Graphics and Text, using AWT controls, Layout managers and menus, handling Image, animation, sound and video, Java Applet. The Collection Framework: The Collection Interface, Collection Classes, Working with Maps &amp; Sets<br>**JDBC:** Introduction to DBMS &amp; RDBMS, DBC API, JDBC Application Architecture, Obtaining a Connection, JDBC Models: Two Tier and Three Tier Model, ResultSet, Prepared Statement, Callable Statement. | |
| **UNIT IV** | **10hrs** |
| **RMI (Remote Method Invocation):** Introduction, Steps in creating a Remote Object, Generating Stub &amp; Skeleton, RMI Architecture, RMI packages. Java Bean: Introduction, Bean Architecture, Using the Bean Development Kit, Creating simple bean-properties, methods and events, Packing beans- the manifest &amp; the jar, Java bean package, Introduction to NetBean.<br>**Swing:** Introduction to JFC (Java Foundation Classes), Features of Swing and Comparison with AWT, Advanced Control in swing (JTree, JTable) . | |
| **Text Books** | |
| 1. Patrick Naughton and HerbertzSchildt, “Java-2: The Complete Reference”, TMH, 2007. | |
| 2. Bill Vanners, “Inside Java Virtual Machine”,TMH, 2 nd Ed, 2000. | |
| 3. Rick Dranell, “HTML 4 unleashed”, Techmedia Publication, 2000. | |
| 4. Paul Dietel and Harvey Deitel, “Java How to Program”, PHI, 8 th Ed., 2010. | |
| **Reference Books** | |
| 1. E. Balaguruswamy, “Programming with Java: A Primer”, TMH, 4 th edition 1998. | |
| 2. N.P Gopalan and J. Akilandeswari, “Web Technology- A Developer’s Perspective”, PHI, 2007. | |

# Data Communications & Networking

| | |
|---|---|
| **Course Code:** MCA-207<br>**Contact Hours:** L-3 P-0<br>**Course Category:** DCC | **Credits:** 4<br>**Semester:** 3 |

**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** Data communications refers to the transmission of this digital data between two or more computers and a computer network or data network is a telecommunications network that allows computers to exchange data. The physical connection between networked computing devices is established using either cable media or wireless media. The best-known computer network is the Internet.

## Course Objectives:

- The students should understand the layers of networking devices.
- They should be familiar with a few networking protocols.
- They should study the different types of networks and topologies of networks.

**Pre-requisite:** Data Structures and Algorithms

**Course Outcomes:** Upon successful completion of this course, students will be able to:

CO1: Describe the fundamental concepts and layered architecture of computer networking.

CO2: Explain the basic concepts of link layer properties to detect error and develop the solution for error control and flow control. Design, calculate, and apply subnet masks and addresses to fulfill networking requirements. Also, compare various routing protocols.

CO3: Comprehend the duties of transport layer and congestion control techniques.

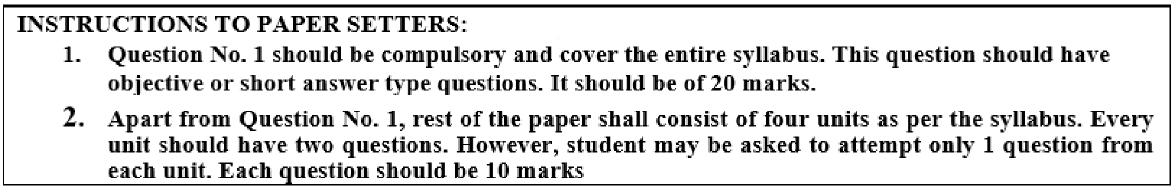
CO4: Illustrate the features and operations of various application layer protocols such as DNS, HTTP, FTP, e-mail protocols and other applications; and focus on network security issues to secure communication towards society.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Students would be encouraged to develop an understanding of the existing real life cyber security issues and how they are solved. Emphasis would be given on assignments where students will be given numerical/ programming assignments based on topics studied in previous lectures. Course will have a blend of theory and practice for the benefit of students. Use of ICT, web-based sources as well as blackboard teaching will be adopted.

| **UNIT-I** | **10 hrs** |
|---|---|
| **Introduction**: Goals and Applications of Networks, Layering Concept, OSI Reference Model vs TCP/IP Protocol Suite, Networks Topology.<br>**Physical Layer:** Signals, Digital Transmission – Analog to Digital & Digital to Digital, Analog Transmission – Digital to Analog & Analog to Analog, Multiplexing – FDM & TDM, Media – Guided and Unguided, Switching – Packet based & Circuit based. Hub & Repeater.<br>**Network Traffic Capturing:** Wireshark (windows) and tcpdump (linux). | |
| **UNIT –II** | **11 hrs** |
| **Data Link Layer:** Addressing; Error Detection & Correction – General concepts, Checksum & CRC; Medium Access – Aloha, CSMA, CSMA/CD & CA; Protocols – Ethernet, ARP & RARP; Switch – Learning & Filtering Mechanism.<br>**Network Layer:** IP Addressing & Subnets; Basic Routing (or Forwarding) Mechanism; IPv4 frame format and functions; Routing protocols – RIP, OSPF & BGP and algorithms – Distance Vector & Link State.<br>**Linux Network Commands**: arp, route, ifconfig, netstat, traceroute, ping. | |
| **UNIT-III** | **11 hrs** |
| **Transport Layer:** Port Addresses; Protocols - Simple, Stop n Wait, Go Back N & Selective Repeat; UDP – Services & Applications; TCP – header format, connection setup & termination, state transition diagram, flow control, error control, congestion control & timers.<br>**Socket Programming:** Socket definition, TCP client & server socket, UDP client & server socket, Problems related to Socket Programming. | |
| **UNIT –IV** | **10 hrs** |
| **Application Layer:** Web & HTTP, FTP, Email, Telnet, SSH, DNS.<br>**Advanced Protocols:** SNMP, RTP, SIP, BitTorrent. | |
| **Text Books** | |
| 1. Forouzan, “Data Communication and Networking”, TMH, 5th Edition, 2013. | |
| 2. A.S. Tanenbaum, “Computer Networks”, PHI, 4th Edition, 2002. | |
| 3. W. Stallings, “Data and Computer Communication”, Macmillan Press, 2013. | |
| 4. Comer, “Computer Networks and Internet”, PHI, 2008 | |
| 5. Comer, “Internetworking with TCP/IP”, PHI, 2008. | |
| **References Books** | |
| 1. W. Stallings, “Data and Computer Communication”, McMillan, 2010 | |
| 2. J. Martin, “Computer Network and Distributed Data Processing”, PHI,2008 | |
| 3. W. Stallings, “Local Networks”, McMillan, 2013. | |
| 4. M.Schwertz, “Computer Communication Network Design and Analysis”, PHI, 1977. | |
| 5. S. Keshav, “An Engineering Approach to Computer Networking, Pearson”, 2001. | |

# Design And Analysis Of Algorithms

| | |
|---|---|
| **Course Code:** MCA-209<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 3 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** Algorithms play a crucial and fundamental role in computer science. Given that algorithms are present in all domains of computer science, it is important for students to learn techniques to analysis a given algorithm. In addition, different approaches to design algorithms are important to write one's own algorithm.

## Course Objectives:

- Introduction, learning and analysis of performances of algorithmic efficiency of approaches such as searching, sorting etc.
- Introduction, learning and analysis of greedy paradigms.
- Introduction, learning and analysis of dynamic programming and back tracking.
- Introduction, learning and analysis of computational complexity and branch & bound.

**Pre-requisites:** Knowledge of data structures and programming.

**Course Outcomes:** After completion of the course, the students will be able:

CO1: Understand asymptotic complexities of the algorithms and design algorithms using Divide and Conquer strategy.

CO2: Apply greedy and dynamic programming approaches for designing algorithms.

CO3: Implement various graph algorithms and design algorithms using backtracking approach and branch and bound techniques

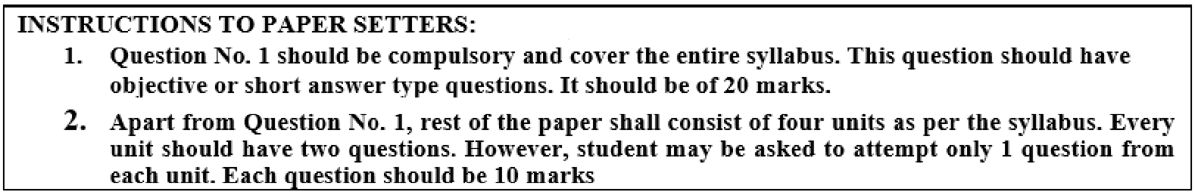
CO4: Implement different string-matching algorithms and understand the concept of NP-complete problems.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web-based resources as well as flipped class room teaching will be adopted.

| **UNIT-I** | **10 hrs** |
|---|---|
| **Introduction to Algorithms**: Need for algorithm, Growth of Functions, Exercises based on Asymptotic Notations, Solving Recurrence Relations–Iterative method, Substitution method & Master method. Space vs Time Complexity Trade-off.<br>**Divide and Conquer Technique:** Merge Sort, Quick Sort, Median and Order Statistics, Maximum-subarray Problem, Strassen's Matrix Multiplication. | |
| **UNIT –II** | **10 hrs** |
| **Dynamic Programming:** Elements of Dynamic Programming, Matrix Chain Multiplication, Longest Common Subsequence, 0/1 Knapsack and Optimal Binary Search Tree problems.<br>**Greedy Algorithms:** Elements of Greedy strategy, Activity Selection problem, Huffman Codes, 0/1 Fractional Knapsack, Task Scheduling problem. | |
| **UNIT-III** | **10 hrs** |
| **Graph Algorithms:** Representation of Graphs, Breadth First Search, Depth First Search, Topological Sort, Strongly Connected Components, Algorithm for Kruskal's and Prim's for finding Minimum cost Spanning Trees, Dijkstra's and Bellman Fort Algorithm for finding Single source shortest paths. All pair shortest paths and matrix multiplication, Floyd Warshall algorithm for all pair shortest paths. | |
| **UNIT –IV** | **10 hrs** |
| **String Matching:** The naïve String-Matching algorithm, The Rabin-Karp Algorithm, String Matching with finite automata, The Knuth-Morris Pratt algorithm.<br>**NP-Completeness:** Polynomial-time verification, NP-Completeness and Reducibility, NP-Completeness Proof, NP-Complete problems. | |
| **Text Books** | |
| 1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C.Stein, "Introduction to Algorithms" PHI, 5th Ed. | |
| 2. Jon Kleinberg and Eva Tardos, "Algorithm Design", Pearson Edition. | |
| **References Books** | |
| 1. Anany Levitin, "Introduction to the Design and Analysis of Algorithm", Pearson Education. | |
| 2. Sara Baase and Allen Van Gelder, "Computer Algorithms - Introduction to Design and Analysis", Pearson Education. | |
| 3. A.V. Aho, J. E. Hopcroft and J.D.Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education. | |
| 4. R. Panneerselvam, "Design and Analysis of Algorithm", PHI. | |

# Computer Graphics And Multimedia Technologies

| | |
|---|---|
| **Course Code:** MCA-202 | **Credits:** 4 |
| **Contact Hours:** L-4 P-0 | **Semester:** 4 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** Computer graphics is an art of drawing pictures, lines, charts, etc. using computers with the help of programming. Computer graphics is made up of number of pixels. Pixel is the smallest graphical picture or unit represented on the computer screen. In this course, students will learn fundamental concept and algorithms of computer graphics and multimedia.

## Course Objectives:

- To learn the fundamental concepts of graphics and multimedia.
- To gain and apply the acquired knowledge pertaining to 2D and 3D concepts in graphics programming.
- To understand the basic 3D modelling and rendering techniques.
- To realize the importance of multimedia towards building the virtual environment and communication.

**Pre-requisites:** Nil

**Course Outcomes**: Upon successful completion of the course, students will be able to:

CO1: Enumerate the functionalities of pixels and coordinate systems pertaining to graphics manipulation.

CO2: Design and demonstrate the 2D and 3D objects using graphics algorithms.

CO3: Have the ability to model and render 3D objects by comprehending the complexities of illumination in virtual scenes.

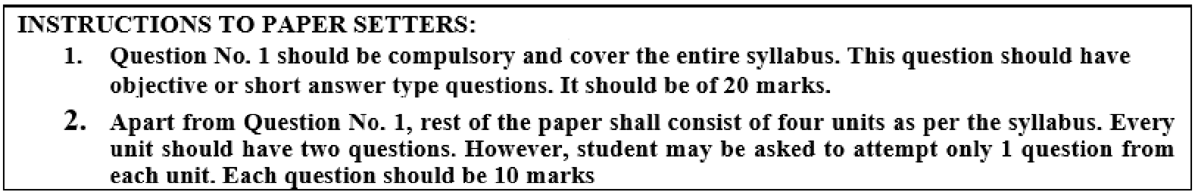
CO4: Appraise and interpret the various multimedia communication standards, applications and basic principles.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web-based resources as well as flipped class room teaching will be adopted.

| | |
|---|---|
| **UNIT I** | **10 Hours** |
| Scan Conversion Algorithms: Scan Converting Lines (DDA, Bresenham), Scan Converting Circles (Mid-point, Bresenham), Scan Converting Ellipses (Midpoint). Clipping: Two-Dimensional Clipping, Sutherland-Cohen Subdivision Line-Clipping Algorithm 2D-Transformation: Representation of Points, Transformations and Matrix, Transformation of Straight Line, 2-D - Rotation, Reflection, Scaling, Combined Transformations, Translation and Homogeneous Coordinates, Translation, Rotation about an Arbitrary Point, Reflection through an Arbitrary Line, window-to-viewport transformation | |
| **UNIT II** | **12 Hours** |
| 3D-Transformation: Representation of Points, 3D- Scaling, 3D- Shearing, 3D- Rotation, Three-Dimensional Translation, 3D- Reflection, Multiple Transformations, Rotation about an Axis Parallel to a Coordinate Axis, Rotation about an Arbitrary Axis in Space. The Dimensional Perspective Geometry: Geometric Projection, Orthographic Projections, Oblique Projections, Perspective Transformations, Single-Point Perspective Transformation, Two-Point Perspective Transformation, Three-Point Perspective Transformation. Solid Modeling: Representing Solids, Regularized Boolean Set Operation primitive Instancing Sweep Representations, Boundary Representations, Spatial Partitioning Representations, Constructive Solid Geometry, Comparison of Representations. | |
| **UNIT III** | **10 Hours** |
| **Representing Curves & Surfaces:** Polygon meshes, parametric, Cubic Curves, geometric and parametric continuities, Hermite, Bezier (4-point, 5-point, general), B-Spline, Quadric Surface Illumination and Shading: Modeling light intensities, ambient light, diffused light, specular reflection, attenuation factor, Reflection vector.<br>**Shading Models:** constant shading, flat shading, gouraud shading, phong shading. Hidden-Surface Removal: Hidden Surfaces and Lines, Back-Face Detection, A-buffer, ZBuffers Algorithm, Scan-line Algorithm, The Painter's Algorithm, Area subdivision Introduction to Multimedia: Multimedia, Multimedia Terms, Introduction to making multimedia – The Stages of project, the requirements to make good multimedia, Multimedia Applications. | |
| **UNIT IV** | **10 Hours** |
| **Multimedia:** Multimedia Hardware, Software and Authoring Tools, Graphics File Formats: TIFF, MIDI, JPEG, MPEG, RTF. Multimedia building blocks – Text, Sound, Images, Animation and Video, Digitization of Audio and Video objects, Data Compression: Different Compression algorithms concern to text, audio, video and images etc. | |
| **Text Books** | |
| 1. Steve Marschner, Peter Shirley, Fundamentals of Computer Graphics, CRC Press, 4th Ed. (2015) | |
| 2. D.Hearn & Baker: Computer Graphics, Prentice Hall of India | |
| 3. Foley, Van Dam, Feiner, Hughes, “Computer Graphics Principles & Practice” | |
| 4. Tay Vaughan, “Multimedia: Making it Work”, TMH | |
| **Reference Books** | |
| 1. K. Andleigh and K. Thakkar, “Multimedia System Design”, PHI, PTR | |
| 2. Rogers & Adams, “Mathematical Elements for Computer Graphics”, McGraw Hill | |

# Business Intelligence

| | |
|---|---|
| **Course Code:** MCA-204<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 4 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

Business Intelligence course provides an overview of business intelligence. The course starts with basic insights into business intelligence and how it differs from data science before covering the key roles and processes involved. You'll learn about each role in detail and what skills are essential in each position. As you progress through the course, you will understand how these roles work together to deliver actionable insights, as well as look at examples of BI tools and how they help revolutionize your work.

## Course Objectives:

- Explain different roles that form part of a business intelligence team
- Discuss how BI serves the needs of a business
- Recognize well and badly designed visuals
- Compare different data types and data structures
- Explain the basics of metrics and functions

## Course Outcomes:

CO1: Understand basic concepts of analysis of data.

CO2: Analyze performance evaluation of classification algorithms.

CO3: Learn and apply classification algorithms.

CO4: Understand unsupervised learning.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web-based resources as well as flipped class room teaching will be adopted.

| | |
|---|---|
| **UNIT I** | **10 Hours** |
| **Introduction**: data mining, different types of learning process, building block with example. Data Exploration and Dimension Reduction: data summaries, data visualization, correlation analysis, Reducing the number of categories in categorical variables, principal component analysis of Breakfast Cereals. | |
| **UNIT II** | **12 Hours** |
| **Evaluation Classification & Predictive Performance**: Introduction, judging classification, accuracy measures, cutoff for classification, performance in unequal importance of classes, asymmetric misclassification costs, oversampling & asymmetric costs, classification using a triage strategy, evaluation predictive performance and some problems. | |
| **UNIT III** | **10 Hours** |
| **Classification and Regression Trees**: introduction, classification trees, recursive portioning, complexity, evaluating the performance of a classification tree, avoiding over fitting: CHAID, pruning the Tree, classification rules from the trees, regression tree, advantages, weaknesses and extensions. | |
| **UNIT IV** | **10 Hours** |
| **Association Rules**: Introduction, transaction database, generates candidate rules, selection of the rules.<br>**Cluster Analysis**: example, Measuring the distance between two records, measuring distance between two clusters, hierarchical clustering, and nonhierarchical clustering.<br>**CASE STUDIES.** | |
| **Text Books** | |
| 1. Galit shmueli,nitin r.patel, peter c.bruce , “Data Mining for Bussiness Intelligence”, Wiley India pvt. Ltd., 2007 | |
| **Reference Books** | |
| 1. Han, J., and Kamber, M. “Data Mining: concepts and techniques.San diego,CA: ACADEMIC Press , 2001. | |
| 2. Hosmer, d. w., and Lemeshow, “Applied Logistic Regression”, New York: Wiey-Interscience, 2nd edition, 2000. | |

# Theory of Computation

| | |
|---|---|
| **Course Code:** MCA-206<br>**Contact Hours:** L-4 P-0 | **Credits:** 4<br>**Semester:** 4 |

**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** The study of automata and the theory of computation deal with the concepts of working of automatic machines and processing of input formal language data. This subject provides an important background material to students involved in understanding the basic functionalities of automata theory.

**Course Objectives:**

- Introduce students to the mathematical foundations of computation including automata theory; the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability.
- Identify different formal language classes and their relationships
- Design grammars and recognizers for different formal languages

**Prerequisite:** Strong background in Discrete mathematics, Data structures and algorithms

**Course Outcomes:** By the end of the course students will be able to:

CO1: Describe the Finite Automata, their capabilities and limitations.

CO2: Classify the different types of grammars, languages and machines.

CO3: Discover the equivalence of languages described by finite state machines and regular expressions.

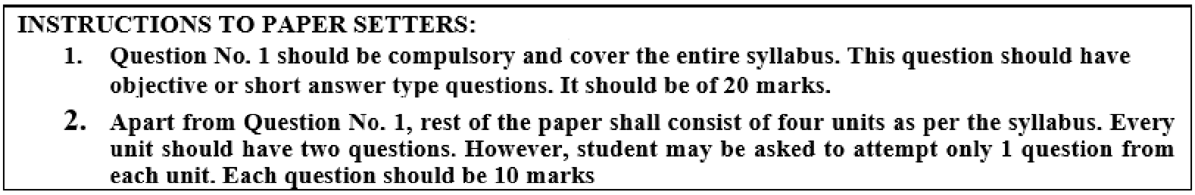
CO4: Design the FA, CFG, Push Down Automata and Turing recognizable languages.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web- based resources as well as flipped classroom teaching will be adopted.

| **UNIT I** | **10 Hours** |
|---|---|
| **Automata and Language Theory**: A brief history of computing, Need of Automata, Overview of Theoretical Computer Science and its application including various phases / Modules in the design of a typical compiler, Chomsky Classification, Introduction to JFLAP Simulation<br>**Finite Automata** : Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Proof of Kleen's Theorem, Arden's Theorem, Myhill Nerode Theorem, Regular Expressions, Transition Graphs, Minimized DFA, Mealy and Moore machines, Equivalence of DFAs, NFAs and Regular Expressions, Closure and Decision properties of Regular Language, Non-Regular Languages, Pumping Lemma., Applications of FA: Text search, Design of lexical analyzer, Finding patterns | |
| **UNIT II** | **10 Hours** |
| **Context Free Grammar and parsers**: CFG, Derivations and Parse trees, Ambiguous Grammar and techniques of removing ambiguity, Chomsky Normal Form, Greibach Normal Form, Closure properties of context free language, Decision problems involving CFLs, Application of CFG: Types of parsers (CYK, Tomita's, LL, LR, SLR), YACC, Mark Up languages<br>**Pushdown Automata** : Deterministic and Non-Deterministic, Equivalence of DPDA, NPDA, CFG and conversion, Language accepted by PDA, Pumping lemma for CFG, Ogden's Lemma | |
| **UNIT III** | **10 Hours** |
| **Linear Bounded Automata**: Power of Linear Bounded Automata, Context Sensitive language, Closure and decision properties<br>**Turing Machines:** Definition**,** General model of computation, TM as language acceptor, enumerator, computing partial functions, Variants and Extension of Turing machine ( One tape, multi tape, Non deterministic, move-in state, stay option etc.), construction of Turing machine, , Church Turing Thesis, Rice's Theorem, halting problem, Hilbert's problem, recursively enumerable language, encoding of Turing machine, L-System | |
| **UNIT IV** | **10 Hours** |
| **Advanced Topics:** Decidability, Reducibility, Computability, Computable functions , recursive, primitive recursive, μ-recursive functions, recursion theorem, post machines, Post Correspondence problem<br>**Time and Space complexity :** P, NP, NP- complete**,** PSAPCE, NPSAPCE, L, NL, EXSPACE, NL- complete, Hierarchy Theorems, Probabilistic Computation, randomness and compressibility (including BPP, ZPP, RP), Zero-Knowledge proof. | |
| **Text Books** | |
| 1. J. C. Martin, "Introduction to Languages and the Theory of Computation", TMH, 3rd Ed., 2007. | |
| 2. M. Sipser, "Introduction to the Theory of Computation", Cengage Publication, 2006. | |
| 3. J. Hopcroft, R. Motwani, and J. Ullman, "Introduction to Automata Theory, Language and Computation", Pearson, 2nd Ed., 2006. | |
| **Reference Books** | |
| 1. H. R. Lewis and C. H. Papadimi Triou, "Elements of the Theory of Computation", Pearson, 2nd Ed., 1997. | |
| 3. Peter Linz, "Introduction to Formal Languages and Automata", Narosa Publishing, 4th ed. 2006. | |

# Cloud Computing

| | |
|---|---|
| **Course Code:** MCA-208<br>**Contact Hours:** L-4 P-0 | **Credits**: 4<br>**Semester:** 4 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** Cloud computing is a scalable service provider platform that provides on-demand and pay per use computing service for various types of shared pool of resources such as memory, servers, storage, networking, software, database, applications designing etc., with the help of the internet. This course will introduce various aspects of cloud computing including fundamentals of cloud computing, load balancing techniques, security challenges, case studies and industrial applications of cloud computing. This will help students to use and explore the cloud computing platforms.

## Course Objectives:

- To learn the use of various cloud computing services and cloud deployment models.
- Understand the concept of virtualization in cloud computing.
- To apply the concepts of cloud computing for designing, evaluating, simulating and comparing various applications in a cloud computing environment.
- To gain the confidence in resource management and load balancing algorithms in a cloud computing environment.
- To gain the confidence of security attacks and their provisions at various levels of cloud computing.

**Prerequisite:** Basic understanding of Operating System.

## Course Outcomes:

CO1: To articulate key concepts of cloud computing and computing techniques, strength and limitations of cloud computing with possible application domains.

CO2: To identify the architecture and infrastructure of cloud computing including SaaS, PaaS, IaaS, public cloud, private cloud and hybrid cloud.

CO3: To interpret various data, scalability and cloud services to acquire efficient database for cloud storage.

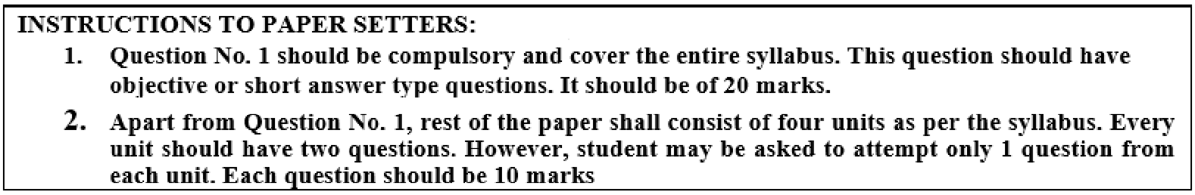
CO4: To explain the core issues of cloud computing such as security, privacy and interoperability and deal with controlling mechanism for accessing sage cloud service.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web-based resources as well as flipped class room teaching will be adopted.

| | |
|---|---|
| **UNIT I** | **10 Hours** |
| **Introduction:** Introduction of cloud computing, History of cloud computing, NIST definition, properties and characteristics, Cloud as green and smart, Cloud as IaaS, PaaS, Saas, BPaaS, HaaS, Public, Private, Hybrid and community cloud, Benefits and Challenges, Applicationavailability, performance, security and disaster recovery; next generation Cloud Applications, Technology providers vs. Cloud providers vs. Cloud vendors. | |
| **UNIT II** | **10 Hours** |
| **Cloud Architecture:** Virtualization concept, cloud building blocks, ROI Model, Service models, deployment models, storage models, security model.<br>**Introduction to IaaS:** Resource Virtualization, Server, Storage, Network **Introduction to PaaS:** Cloud platform & Management, Computation, Storage**Introduction to SaaS:** Web services, Web 2.0, Web OS.<br>**Cloud Storage Infrastructure:** Storage strategy and governance; security and regulations Storage Network Design: Architecture of storage, analysis and planning. Storage network designconsiderations, Cloud Optimized Storage, Designing backup/recovery solutions. | |
| **UNIT III** | **10 Hours** |
| **Cloud issues and challenges:** Cloud provider Lock-in, Security challenges and approaches(Infrastructure security, Network level security, Host level security, Application levelsecurity,Data security and Storage, Data privacy and security Issues, Jurisdictional issues raisedby Data location, Identity & Access Management, Access Control, Trust, Reputation, Risk Authentication in cloud computing, Client access in cloud, Cloud contracting Model, Commercial and business considerations. | |
| **UNIT IV** | **10 Hours** |
| **Application Development**: Service creation environments to develop cloud-based applications, Development environments for service development; Amazon, Azure, Google App, Salesforce.com, IBM Cloud, Google MapReduce, Yahoo Hadoop, Eucalyptus, Nimbus, OpenStack. | |
| **Text Books** | |
| 1. Barrie Sosinsky, "Cloud Computing Bible", Wiley-India 1st edition, 2011 | |
| 2. Gautam Shroff, "Enterprise Cloud Computing Technology Architecture Applications"Cambridge University Press 1st edition, 2010 | |
| 3. Greg Schulz, "Cloud and Virtual Data Storage Networking", Auerbach Publications, 1stedition, 2009 | |
| **Reference Books** | |
| 1. Miller Michael, "Cloud Computing: Web-Based Applications That Change the WayYou Work and Collaborate Online", Pearson Education India ,1st edition, 2008, | |
| 2. Ronald L. Krutz, Russell Dean Vines, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", Wiley-India 1st edition, 2010 | |

# Advanced Database Management Systems

| | |
|---|---|
| **Course Code:** MCA-301 | **Credits:** 4 |
| **Contact Hours:** L-4 P-0 | **Semester:** 5 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** This course will help the students to sharpen their DBMS skills in more depth. This course describes in major details about the advanced concepts of database management systems including advanced SQL, handling unstructured data, Query execution, database security and various database models.

## Course Objectives:

- To sharpen the skills on writing complex and effective queries
- To handle unstructured data by using No-SQL and MongoDB
- To understand the query execution plan
- To design and implement Distributed Databases.

**Prerequisite:** Basic DBMS concepts and any Programming Language.

**Course Outcomes:** At the end of the course, students will be able to:

CO1: Understand and use the unstructured big data along with concepts like Hadoop, Map Reduce, NoSQL, Pig and Hive for management and analytics.

CO2: Implement various advanced concepts of Database management Systems like Object Oriented System, Distributed Database Systems and Multimedia Database Management Systems for database design.

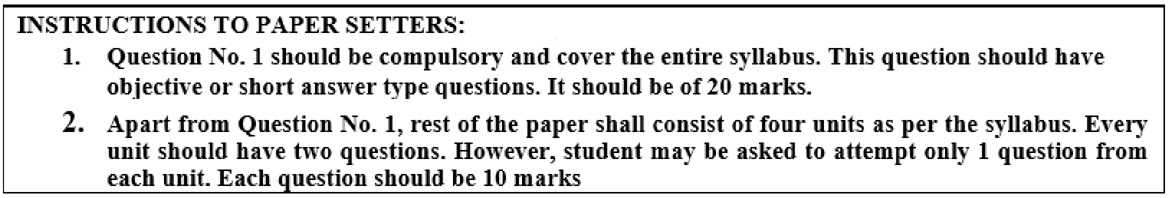
CO3: Write appropriate programs (Procedures/Functions/Triggers) at Server side for better, efficient and secure application development.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web- based resources as well as flipped classroom teaching will be adopted.

| | |
|---|---|
| **UNIT 1** | **10 hrs** |
| Fundamentals of Relational Model, Advanced SQL queries: Nested Queries, Joins, Correlated Queries, Views, Indexes, Sequence. PL/SQL: Exceptions, Cursors, Triggers, Functions, Procedures, Packages. | |
| **UNIT II** | **11 hrs** |
| Indexing &Hashing, B+ Tree Index Files, B-Tree Index Files, Dynamic & Static Hashing, Query Processing, Measures of Query cost, Selection Operation, Sorting, Join operation, evaluation of expressions, Query Optimization, Estimating Statistics of Expression.<br>Results, Transformation of Relational Expressions, Materialized Views | |
| **UNIT III** | **11 hrs** |
| Object Oriented and Object Relational Database Systems, Abstract Data Types, Varying Array, Nested Tables.<br>Distributed Databases, Homogeneous & Heterogeneous Databases, Distributed Data Storage, Distributed Transactions and their commit protocols, Concurrency Control in Distributed Data Bases, Decision Support Systems, Multimedia Databases, Mobile Data bases, Spatial Database. | |
| **UNIT IV** | **10 hrs** |
| Big Data-Volume, Velocity, Variety, Veracity, Types and Sources of Big Data OLAP & RTAP, Data Exploration, Data Summaries, Data Visualization, Tools for Big Data<br>Analytics, No SQL, Hadoop, Map Reduce, Gephi | |
| **Text Books** | |
| 1. Fundamentals of Database System, by Elmasri Ramez and Navathe Shamkant, Pearson, 7th Edition, 2017 | |
| 2. Big Data Analytics, Radha Shankarmani and M. Vijayalakshmi, Wiley, 2nd Edition 2016 | |
| **Reference Books** | |
| 1. Database System Concepts, by Abraham Silberschatz and Hank Korth, McGraw Hill Publication, 6th Edition, 2013 | |
| 2. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, by Davy Cielen and Arno D.B. Meysman, Dreamtech Publication, 2016 | |

# Software Testing and Quality Assurance

| | |
|---|---|
| Course Code: MCA-303<br>Contact Hours: L-4 P-0 | Credits: 4<br>Semester: 5 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

**Introduction:** This course introduces concepts, metrics, and models in software testing and quality assurance. The course covers components of software quality assurance systems before, during, and after software development. It presents a framework for software quality assurance and discuss individual components in the framework such as planning, reviews, testing, configuration management, and so on. It also discusses metrics and models for software quality as a product, in process, and in maintenance. The course will include case studies and hands on experiences. Students will develop an understanding of software quality and approaches to assure software quality.

## Course Objective:

- To understand that software testing is a fundamental part of the software life cycle.
- To learn the essential theories, types, tools, and methods of software testing
- To learn about various software testing problems.
- Understand the basic tenets of software quality and quality factors.
- Be exposed to the Software Quality Assurance (SQA) architecture and the details of SQA components.
- Understand of how the SQA components can be integrated into the project life cycle.
- Be familiar with the software quality infrastructure.
- Be exposed to the management components of software quality.

**Prerequisite:** General knowledge of Software Engineering and Software development life cycle.

**Course Outcomes:** After the completion of the course, the student will be able to

CO1: Understand the fundamental concepts of a software testing and software quality assurance.

CO2: Derive test cases using different testing strategies.

CO3: Generate and prioritize test cases to prove the correctness of program implementations and understand testing at different levels

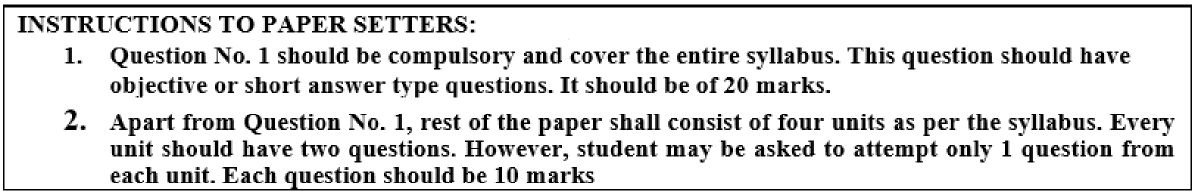
CO4: Understand software Quality Assurance methods, models and measurement.

**Pedagogy:** The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Faculty members strive to make the classes interactive so that students can correlate the theories with practical examples for better understanding. Use of ICT, web- based resources as well as flipped classroom teaching will be adopted.

| | |
|---|---|
| **UNIT I** | **10 hours** |
| **Introduction to Software Testing:** Testing as an Engineering Activity, Testing Fundamentals, Software Testing Process, Software Testing principles, Defects-Hypothesis and Tests, Test Strategy, Test Plan, Software Testing Tools<br>**Software Quality**: Software Quality Fundamentals, Software Quality Management Process, Practical Considerations, Software Quality Tools | |
| **UNIT II** | **10 hours** |
| **Testing Techniques**: Levels of Testing, Functional Testing: Boundary value analysis, Equivalence partitioning, Decision table, White Box Testing: Static testing techniques, Static analysis tools, Control flow testing, Code complexity testing, Data flow testing, Tools for software testing | |
| **UNIT III** | **10 hours** |
| **Integration, System and Acceptance Testing**: Integration testing approaches, System testing, Non- functional testing techniques, Acceptance Testing, **Fault based testing:** Regression testing, Regression test process, Regression, Mutation Testing, Test Minimization, Software Test Automation | |
| **UNIT IV** | **10 hours** |
| **Software Quality Assurance:** Software Quality, Software Quality Indicators, Concepts of Quality Control, Garvin's Quality Dimensions, McCall's Quality Factors, Software Quality Dilemma, Achieving Software Quality, Elements of Software Quality Assurance, Software Quality Assurance Metrics, Quality Assurance Models, Total Quality Management, Software Quality Assurance Plan | |
| **Text Books** | |
| 1. Yogesh Singh, "Software Testing", Cambridge University Press, 2013/ Latest Edition. | |
| 2. 2.P Paul C. Jorgensen, "Software Testing: A Craftsman's Approach", Auerbach Publications,4th Edition, 2013/ Latest Edition. | |
| **Reference Books/Materials** | |
| 1. Burnstein, "Practical Software Testing: A Process-Oriented Approach", Springer, 3rd Edition**,** 2003/ Latest Edition. | |
| 2. Aditya P. Mathur, "Foundations of Software Testing", Pearson, 2nd Edition, 2013/ Latest Edition. | |
| 3. S. Naik, P. Tripathy," Software Testing and Quality Assurance", Wiley, 2010/ Latest Edition. | |
| 4. Milind Limaye, "Software Quality Assurance", McGraw-Hill publication, 2011/ Latest Edition. | |
| 5. https://nptel.ac.in/courses/106/101/106101163/ | |

# Network Security

| | |
|---|---|
| **Course Code:** MCA-305<br>**Contact Hours:** L- 4 P-0 | **Credits:** 4<br>**Semester:** 5 |



**INSTRUCTIONS TO PAPER SETTERS:**

1. **Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 20 marks.**
2. **Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks**

## Introduction:

This course will introduce students to the basic building blocks of cryptography and applications of cryptographic protocols in real world and network security. The intent of this course is to familiarize students with security threats, cryptography, and application development in computer network protocols. The focus will be on how cryptography and its applications can maintain privacy and security in electronic communications and computer networks.

## Course Objective:

- To understand the fundamentals of Cryptography.
- To acquire knowledge on standard algorithms used to provide confidentiality, integrity and authenticity.
- To explain and use modern cryptographic methods (symmetric encryption, public key encryption, hash functions, key management, digital signatures, certificates etc).
- To discuss various network security protocols.

**Pre-requisite:** None

**Course Outcome:** Upon successful completion of this course, students will be able to:

CO1: Understand network security basics.

CO2: Analyze and differentiate between public-key and private key cryptosystems.

CO3: Evaluate security mechanisms using rigorous approaches by key ciphers and hash functions.

CO4: Design cryptographic protocols to solve real world problems.

## Pedagogy:

The teaching-learning of the course would be organized through lectures, tutorials, assignments, projects/ presentations and quizzes. Students would be encouraged to develop an understanding of the existing real life network security issues and how they are solved. Emphasis would be given on assignments where students will be given numerical/ programming assignments based on topics studied in previous lectures. Course will have a blend of theory and practice for the benefit of students. Use of ICT, web based sources as well as blackboard teaching will be adopted.

| **UNIT I** | **10 hours** |
|---|---|
| Introduction and terminology, Conventional Cryptography: Definitions, Classical encryption techniques, Substitution and Transposition Cipher, Vignere Cipher, Introduction to security attacks, services and mechanism, Security Overview, CIA model, Security Policies and Mechanisms, Threats, Block Ciphers and Stream Ciphers, Block ciphers principles, Shannon's theory of confusion and diffusion, Fiestal Structure, Data Encryption Standard (DES), Cryptanalysis of DES, Triple DES. | |
| **UNIT II** | **10 hours** |
| Group, Abelian and Cyclic group, Ring, Finite Fields Advanced Encryption Standard (AES), Modes of Encryption: ECB, CBC, CFB, Counter mode, Message Padding, Asymmetric Cryptography: Number Theory, Modular Arithmetic, Fermat's and Euler's theorem, primarily testing, Euclid's Algorithm, Chinese Remainder theorem, discrete logarithms, public key cryptography: RSA, El Gamal, and Elliptic Curve Cryptography, Diffie Hellman Key management, Meet-in-the-Middle Attack, Digital Certificates: X.509. | |
| **UNIT III** | **10 hours** |
| Digital Signatures, Stream Ciphers, LFSR based stream ciphers, Hash functions, Hash algorithms (MD5, SHA-2, Kecchak), Message Authentication Codes, CBC-MAC, HMAC, NMAC, Authentication Protocols: Kerberos, password, challenge-response, biometric authentication, electronic mail security-pretty good privacy (PGP), S/MIME, Malicious Logic, Trojan Horses, Defenses, Viruses, Worms Logic Bombs, Sandboxing. | |
| **UNIT IV** | **10 hours** |
| IP Security: Architecture, Authentication header, encapsulating security payloads, combining security associations, key management, Web Security: Secure Socket Layer (SSL) and transport layer security, TSP, Secure Electronic Transaction (SET), Electronic money, firewall design principals, Virtual Private Network (VPN) security. | |
| **Text Books** | |
| 1. W. Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, 7th Edition, 2017/Latest edition. | |
| 2. B. Forouzan, D. Mukhopadhyay, "Cryptography and Network Security", McGrawHill Education, 3rd Edition, 2015/Latest edition | |
| **Reference Books** | |
| 1. A. Menezes, P. Oorschot, S. Vanstone, "Handbook of Applied Cryptography", Hardcover Edition, CRC press, 2018/Latest edition. | |
| 2. R. Stinson, M. Paterson, "Cryptography: Theory and Practice", CRC Press, 4th Edition, 2018/Latest edition. | |
| 3. B. Menezes, "Network Security and Cryptography", Delmar Cengage Learning, 2nd Edition, 2012/Latest edition | |
| 4. M. Bishop, "Introduction to Computer Security", Addison-Wesley Professional, 3rd Edition, 2005/Latest edition. | |