# Scheme of Examinations and Syllabi
# for
# M.Sc. Computer Science Program

**(Based on Curriculum and Credit Framework for PG Programs under NEP-2020)**

**With Effect from the Session 2024-25**

## MAHARSHI DAYANAND UNIVERSITY
## ROHTAK (HARYANA)

## Structure for 2-year M.Sc. Computer Science Program

| | Semester | Discipline-Specific Courses (DSC) | Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/ Internship | Research thesis/Project | Total Credits |
|---|---|---|---|---|---|
| **First year of 2 Year PG program (NHEQF Level 6)** | | | | | |
| | I | Discrete Mathematical Structures @ 4 credits | Office Automation Tools/Web Development-I / Internship 1 @ 4 credits | --- | 24 |
| | | Computer Networks @ 4 credits | | | |
| | | Computer Fundamentals and Programming in C @ 4 credits | | | |
| | | Computer Organization and Architecture @ 4 credits | | | |
| | | Database Management Systems@ 4 credits | | | |
| | II | Data Structures using C @ 4 credits | Multimedia and Animation / Web Development-II / Internship 2 @ 4 credits | --- | 24 |
| | | Operating Systems @ 4 credits | | | |
| | | Object Oriented Programming using C++ @ 4 credits | | | |
| | | Artificial Intelligence @ 4 credits | | | |
| | | Software Engineering @ 4 credits | | | |
| **Students who exit after first year on completion of 48 credits will be awarded PG Diploma in concerned discipline** | | | | | |
| **Second year of two year PG program (NHEQF Level 6.5)**<br>**(STUDENT SHOULD SELECT ANY ONE OPTION FOR THE SECOND YEAR OF 2 YEAR PG PROGRAM)** | | | | | |
| **Only Course Work** | | | | | |
| Option 1 | III | Design and Analysis of Algorithms@ 4 credits | Full Stack Development-I / Internship 3<br>@ 4 credits | --- | 24 |
| | | Java Programming@ 4 credits | | | |
| | | Computer Security@ 4 credits | | | |
| | | Data Warehousing & Mining@ 4 credits | | | |
| | | Computer Graphics @ 4 credits | | | |
| | IV | Cloud Computing @ 4 credits | Full Stack Development-II /Internship 4 @ 4 credits | --- | 24 |
| | | Mobile Application Development using | | | |

| | | | | | |
|---|---|---|---|---|---|
| | | Android @ 4 credits<br>Internet of Things@ 4 credits<br>Data Visualization & Interpretation @ 4 credits<br>Compiler Design @ 4 credits | | | |
| **Course work and Research** | | | | | |
| **Option 2** | **III** | **Design and Analysis of Algorithms @ 4 credits**<br>**Java Programming @ 4 credits**<br>**Computer Security @ 4 credits**<br>**Data Warehousing & Mining @ 4 credits**<br>**Computer Graphics @ 4 credits** | **Full Stack Development-I / Internship 3 @ 4 credits** | **---** | **24** |
| | **IV** | **--** | **Relevance of Research in Computer Science/Internship 4 @ 4 credits** | **Research thesis@20 credits** | **24** |
| **Only Research (only for the students who have completed 3 Years Bachelor's Program)** | | | | | |
| | **Semester** | **Discipline-Specific Courses (DSC)** | **Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/ Internship** | **Research thesis/project** | **Total Credits** |
| **Option 3** | **III** | **--** | **Probability and Statistics for Computer Science/Internship 3 @ 4 credits** | **20 credits*** | **24** |
| | **IV** | **--** | **Relevance of Research in Computer Science/Internship 4 @ 4 credits** | **20 credits**** | **24** |

**Note:**

***The students who opted Option 3 should submit a project report/synopsis of atleast 50 pages comprising of Literature survey, identification of Research Problem, Plan of work, methodology as well as practical work (if any) at the end of 3rd semester and the same will be evaluated by internal and external examiners.

**The students should continue the research work in 4thsemester based on the project work/synopsis submitted at the end of 3rdsemester. The final thesis/project report will be evaluated by the internal and external examiners.

## Structure for 1 year M.Sc. Computer Science Program (2nd year of 2 Year PG Program)

| | Semester | Discipline-Specific Courses (DSC) | Skill Enhancement Courses (SEC) / Vocational Courses (VOC)/Internship | Dissertation/ Project work | Total Credits |
|---|---|---|---|---|---|
| **(STUDENT SHOULD SELECT ANY ONE OPTION)** | | | | | |
| **Only Course Work** | | | | | |
| **Option 1** | **I (Semester III of 2 year PG Program)** | **Artificial Intelligence @ 4 credits** | **Internship 3 @ 4 credits** | **---** | **24** |
| | | **Java Programming @ 4 credits** | | | |
| | | **Information Security and Privacy @ 4 credits** | | | |
| | | **Data Warehousing & Mining @ 4 credits** | | | |
| | | **Operation Research @ 4 credits** | | | |
| | **II (Semester III of 2 year PG Program)** | **Wireless Technologies @ 4 credits** | **Internship 4 @ 4 credits** | **---** | **24** |
| | | **Cyber Security @ 4 credits** | | | |
| | | **Data Visualization & Interpretation @ 4 credits** | | | |
| | | **Cloud Computing @ 4 credits** | | | |
| | | **Social Network Analysis @ 4 credits** | | | |
| **Course work and Research** | | | | | |
| **Option 2** | **I (Semester III of 2 year PG Program)** | **Artificial Intelligence @ 4 credits** | **Internship 3 @ 4 credits** | **---** | **24** |
| | | **Java Programming @ 4 credits** | | | |
| | | **Information Security and Privacy @ 4 credits** | | | |
| | | **Data Warehousing & Mining @ 4 credits** | | | |
| | | **Operation Research @ 4 credits** | | | |
| | **II (Semester IV of 2 year PG Program)** | **--** | **Relevance of Research in Computer Science/Internship 4 @ 4 credits** | **Research Thesis @ 20 credits** | **24** |

| Type of Course | | | Credits Distribution | | | Total Credits | Workload | | | Total Workload | Marks | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Nomenclature of Course | Course Code | L | T | P | | L | T | P | | Theory | | Practical | | Total Marks |
| | | | | | | | | | | | Internal | External | Internal | External | |
| Semester I (Session 2024-25) | | | | | | | | | | | | | | | |
| **DSC1 @ 4 credits** | Discrete Mathematical Structures | 24CSC201DS01 | 4 | 0 | 0 | 4 | 4 | 0 | 0 | 4 | 30 | 70 | 0 | 0 | 100 |
| **DSC2 @ 4 credits** | Computer Networks | 24CSC201DS02 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC3 @ 4 credits** | Computer Fundamentals and Programming in C | 24CSC201DS03 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC4 @ 4 credits** | Computer Organization and Architecture | 24CSC201DS04 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC5 @ 4 credits** | Database Management Systems | 24CSC201DS05 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **SEC1/VOC1/Internship I @ 4 credits** | Office Automation Tools | 24CSC201SE01 | 2 | 0 | 2 | 4 | 2 | 0 | 4 | 6 | 15 | 35 | 15 | 35 | 100 |
| | OR | OR | | | | | | | | | | | | | |
| | Web Development-I | 24CSC201MV01 | 2 | 0 | 2 | 4 | 2 | 0 | 4 | 6 | 15 | 35 | 15 | 35 | 100 |
| | OR | OR | | | | | | | | | | | | | |
| | Internship- I | 24CSC201IN01 | 0 | 0 | 4 | 4 | 0 | 0 | 8 | 8 | | | 30 | 70 | 100 |
| Semester II (Session 2024-25) | | | | | | | | | | | | | | | |
| **DSC 6 @ 4 credits** | Data Structures using C | 24CSC202DS01 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 7 @ 4 credits** | Operating Systems | 24CSC202DS02 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 8 @ 4 credits** | Object Oriented Programming using C++ | 24CSC202DS03 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 9 @ 4 credits** | Artificial Intelligence | 24CSC202DS04 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 10 @ 4 credits** | Software Engineering | 24CSC202DS05 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **SEC2 / VOC2/ Internship II @ 4 credits** | Multimedia and Animation | 24CSC202SE01 | 2 | 0 | 2 | 4 | 2 | 0 | 4 | 6 | 15 | 35 | 15 | 35 | 100 |
| | OR | OR | | | | | | | | | | | | | |
| | Web Development-II | 24CSC202MV01 | 2 | 0 | 2 | 4 | 2 | 0 | 4 | 6 | 15 | 35 | 15 | 35 | 100 |
| | OR | OR | | | | | | | | | | | | | |
| | Internship – II | 24CSC202IN01 | 0 | 0 | 4 | 4 | 0 | 0 | 8 | 8 | 0 | 0 | 30 | 70 | 100 |

| **Type of Course** | | | **Credits Distribution** | | | **Total Credits** | **Workload** | | | **Total Workload** | **Marks** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Nomenclature of Course** | **Course Code** | **L** | **T** | **P** | | **L** | **T** | **P** | | **Theory** | | **Practical** | | **Total Marks** |
| | | | | | | | | | | | **Internal** | **External** | **Internal** | **External** | |
| Semester III (Session 2025-26) **OPTION 1/2** | | | | | | | | | | | | | | | |
| **DSC 11 @ 4 credits** | Design and Analysis of Algorithms | 25CSC203DS01 | 4 | 0 | 0 | 4 | 4 | 0 | 0 | 4 | 30 | 70 | 0 | 0 | 100 |
| **DSC 12 @ 4 credits** | Java Programming | 25CSC203DS02 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 13 @ 4 credits** | Computer Security | 25CSC203DS03 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 14 @ 4 credits** | Data Warehousing & Mining | 25CSC203DS04 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 15 @ 4 credits** | Computer Graphics | 25CSC203DS05 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **SEC3/ Internship III/ Project Work 1@ 4 credits** | Full Stack Development-I | 25CSC203SE01 | 2 | 0 | 2 | 4 | 2 | 0 | 4 | 6 | 15 | 35 | 15 | 35 | 100 |
| | OR | OR | | | | | | | | | | | | | |
| | Internship –III | 25CSC203IN01 | 0 | 0 | 4 | 4 | 0 | 0 | 8 | 8 | 0 | 0 | 30 | 70 | 100 |
| Semester III (Session 2025-26) **OPTION 3** | | | | | | | | | | | | | | | |
| **Skill Enhancement Courses (SEC) /Vocational Courses (VOC)/ Internship III @ 4 credits** | Probability and Statistics for Computer Science | 25CSC203SE02 | 4 | 0 | 0 | 4 | 4 | 0 | 0 | 4 | 30 | 70 | 0 | 0 | 100 |
| | OR | OR | | | | | | | | | | | | | |
| | Internship-III | 25CSC203IN02 | 0 | 0 | 4 | 4 | 0 | 0 | 8 | 8 | 0 | 0 | 30 | 70 | 100 |
| **Research thesis/project @20 credits** | Research Thesis-1 | 25CSC203PD01 | - | - | - | 20 | - | - | - | - | - | - | 150 | 350 | 500 |
| Semester IV (Session 2025-26) **OPTION 1** | | | | | | | | | | | | | | | |
| **DSC 16 @ 4 credits** | Cloud Computing | 25CSC204DS01 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC 17 @ 4 credits** | Mobile Application Development using Android | 25CSC204DS02 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC18 @ 4 credits** | Internet of Things | 25CSC204DS03 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC19 @ 4 credits** | Data Visualization & Interpretation | 25CSC204DS04 | 3 | 0 | 1 | 4 | 3 | 0 | 2 | 5 | 25 | 50 | 5 | 20 | 100 |
| **DSC20 @ 4 credits** | Compiler Design | 25CSC204DS05 | 4 | 0 | 0 | 4 | 4 | 0 | 0 | 4 | 30 | 70 | 0 | 0 | 100 |
| **SEC4/Internship IV/** | Full Stack | 25CSC204SE01 | 2 | 0 | 2 | 4 | 2 | 0 | 4 | 6 | 15 | 35 | 15 | 35 | 100 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Project Work 2 @ 4 credits** | Development-II | | | | | | | | | | | | | | |
| | OR | OR | | | | | | | | | | | | | |
| | Internship-IV | 25CSC204IN01 | 0 | 0 | 4 | 4 | 0 | 0 | 8 | 8 | 0 | 0 | 30 | 70 | 100 |
| Semester IV (Session 2025-26) **OPTION 2/3** | | | | | | | | | | | | | | | |
| **SEC4/Internship IV @ 4 credits** | Relevance of Research in Computer Science | 25CSC204SE01 | 4 | 0 | 0 | 4 | 4 | 0 | 0 | 4 | 30 | 70 | 0 | 0 | 100 |
| | OR | OR | | | | | | | | | | | | | |
| | Internship –IV | 25CSC204IN01 | 0 | 0 | 4 | 4 | 0 | 0 | 8 | 8 | 0 | 0 | 30 | 70 | 100 |
| **Research thesis/ project @20 credits** | Research Thesis-II | 25CSC204PD01 | - | - | - | 20 | - | - | - | - | - | - | 150 | 350 | 500 |

**L: Lecture; T: Tutorial; P: Practical**

# Program Learning Outcomes of M.Sc. Computer Science Program

The graduate on completion of M.Sc. Computer Science program will be able to:-

| | |
|---|---|
| PLO1 | Acquire advanced knowledge and a critical understanding of emerging developments and issues relating to the domain of computer science as life-long skills. |
| PLO2 | Exhibit an advanced understanding of the principles, methods, and techniques applicable to theoretical and practical contexts. |
| PLO3 | Extrapolate acquired knowledge and skills to real-life situations, new and unfamiliar contexts, for problem solving and create innovative solutions. |
| PLO4 | Demonstrate the ability to apply conceptual, operational, and technical knowledge with a range of cognitive and practical skills. |
| PLO5 | Apply advanced knowledge of research methods to conduct research and investigations to formulate evidence-based solutions using a systematic problem solving approach for complex problems that require higher order thinking. |
| PLO6 | Communicate their solutions, application, research findings and professional insights in a well-structured and coherent manner to both specialist and non-specialist audiences. |
| PLO7 | Engage in self-directed learning to continually upgrade their knowledge and skills, including research competencies along with ethical standards and practices in their professional and academic endeavors for a life-long learning. |

# Syllabi for M.Sc. Computer Science Program

## Semester: First
## Session: 2024-2025

| | | | |
|---|---|---|---|
| **Name of Program** | **M.Sc. Computer Science** | **Program Code** | **-----------** |
| **Name of the Course** | **Discrete Mathematical Structures** | **Course Code** | **24CSC201DS01** |
| **Hours per Week** | **4+0+0** | **Credits** | **4:0:0** |
| **Maximum Marks** | **Theory: 100(70+30)** | **Time of Examinations** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**
The objective of this course is to provide students with a solid foundation in discrete mathematics, which is essential for the study of computer science and related fields. Students will learn about set theory, relations, functions, predicate calculus, Boolean algebra, logics, graph theory, related algorithms, formal languages and automata theory. The course will emphasize problem-solving skills and the application of mathematical concepts to computer science.

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

CO1: Understand and apply the basic concepts of set theory.
CO2: Comprehend and use propositions, logical operators, and expressions.
CO3: Understand the use of quantifiers and inference for propositions and predicates.
CO4: Understand the fundamental concepts, laws, and rules of Boolean algebra
CO5: Understand the concept of formal languages and automata theory.

**Unit-I**

**Set Theory:** Definition of sets, countable and uncountable sets, Venn Diagrams, proofs of some general identities on sets.
**Relation:** Definition, types of relation, composition of relations, Pictorial representation of relation, equivalence relation, partial ordering relation.
**Function:** Definition, type of functions, one to one, into and onto function, inverse function, composition of functions, Recursive Functions.

**Unit-II**

**Propositional Logic:** Proposition logic, basic logic, Logical Connectives, truth tables, tautologies, contradiction, Logical implication, Logical equivalence, Normal forms, Theory of Inference and deduction.
**Predicate Calculus:** Predicates and quantifiers. Mathematical Induction.

**Unit-III**

**Matrices:** Definition, Types of Matrices, Addition, Subtraction, Scalar Multiplication and Multiplication of Matrices, Adjoint and Inverse of a matrix.
**Determinants:** Definition, Minors, Cofactors, Properties of Determinants, Applications of determinants in finding area of triangle, Solving a system of linear equations.

**Unit-IV**

**Formal Languages:** Introduction to defining language, Kleene Closure, Arithmetic expressions, Chomsky Hierarchy, Regular expressions.
**Automata Theory:** Conversion of regular expression to Finite Automata, NFA, DFA, Conversion of NFA to DFA, FA with output: Moore machine, Mealy machine.

**Suggested Readings:**

1. C.L. Liu, Elements of Discrete Mathematics, McGraw-Hill.
2. S. Lipschutz, Discrete Mathematics, Schaum's Outline Series, McGraw-Hill.
3. B. Ram, Discrete Mathematics, Vinayek Publishers, New Delhi.
4. J.P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, TMH.
5. K.H. Rosen, Discrete Mathematics and Its Applications, TMH.
6. A. Doerr and K. Levasseur, Applied Discrete Structures for Computer Science, Galgotia Publications Pvt. Ltd.
7. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Computer Networks** | **Course Code** | **24CSC201DS02** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**

The objective of this course is to gain a fundamental understanding of computer networks, including their design, protocols, and operation. This will enable students to analyze network communication processes, troubleshoot network issues, and configure basic network devices.

**Course Outcomes:**

By the end of the course the students will be able to:

CO1: Independently understand basic computer network technology.
CO2: Understand and explain Data Communications System and its components, different types of network topologies and protocols.
CO3: Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer, different types of network devices and their functions within a network.
CO4: Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.
CO5: Understanding High-Speed LANs and concept of Network Security.

## Unit-I

**Introduction to Computer Network:** Types of Networks, Network Topologies, OSI and TCP/IP Reference Models;

**Data Communications Concepts:** Digital Vs. Analog communication; Parallel and Serial Communication; Synchronous, Asynchronous and Isochronous Communication; Communication modes: simplex, half duplex, full duplex; Multiplexing;

**Transmission media:** Wired-Twisted pair, Coaxial cable, Optical Fibre, Wireless transmission: Terrestrial, Microwave, Satellite, and Infrared.

## Unit-II

**Communication Switching Techniques:** Circuit Switching, Message Switching, Packet Switching.

**Data Link Layer Fundamentals:** Framing, Basics of Error Detection, Forward Error Correction, Cyclic Redundancy Check codes for Error Detection, Flow Control. Media Access Protocols: ALOHA, Carrier Sense Multiple Access (CSMA), CSMA with Collision Detection (CSMA/CD), Token Ring, Token Bus.

## Unit-III

**High-Speed LAN:** Standard Ethernet, Fast Ethernet, Gigabit Ethernet, 10G; Wireless LANs: IEEE 802.11, Bluetooth.

**Network Layer:** IP Addressing and Routing, Network Layer Protocols: IPv4 (Header Format and Services), ARP, ICMP (Error Reporting and Query message); IPv6 (Header Format and Addressing).

## Unit-IV

**Transport Layer:** Process-to-Process Delivery: UDP, TCP; Application Layer: Domain Name System (DNS); SMTP; HTTP; WWW.

**Network Security:** Security Requirements and attacks; Cryptography: Symmetric Key (DES, AES), Public Key Cryptography (RSA); Firewall.

**Suggested Readings:**

1. B.A. Forouzan, Data Communications and Networking, McGraw-Hill.
2. W. Stallings, High-Speed Networks and Internets: Performance and Quality of Service, Pearson Education.
3. Gill, Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing Company (Pvt.) Limited, New Delhi.
4. D.E. Comer, Internetworking with TCP/IP: Volume I, Principles, Protocols, and Architectures, Pearson Education.
5. G. Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design, Pearson Education.
6. Gill, Nasib Singh: Essentials of Computer and Network Technology, Khanna Books Publishing Co.(P), New Delhi.
7. B. Muthukumaran, Introduction to High Performance Networks, Vijay Nicole Imprints.
8. W. Tomasi, Introduction to Data Communications and Networking, Pearson Education.
9. J.F. Kurose and K.W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, Pearson Education.
10. A.S. Tanenbaum, Computer Networks, Pearson Education.
11. M. Hassan and R. Jain, High Performance TCP/IP Networking: Concepts, Issues, and Solutions, Pearson Education.
12. A.S. Tanenbaum and M. Van Steen, Distributed Systems: Principles & Paradigms, Pearson Education.
13. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of programs/Lab Assignments**

1. Write a program to simulate Bus topology
2. Write a program to simulate Star topology.
3. Write a program to simulate Ring topology.
4. Write a program to simulate Mesh topology.
5. Write a program to demonstrate the difference between digital and analog signals using simple sine waves and square waves.
6. Install and Configure Wired and Wireless NIC.
7. Connect the computers in Local Area Network.
8. Program to demonstrate simplex: half-duplex communication using text-based examples.
9. Program to demonstrate simplex: full-duplex
10. Install and configure Network Devices: Routers.
11. Configure Host IP, Subnet Mask and Default
12. Program to simulate a basic circuit switching network where a connection is established before communication.
13. Program to simulate packet switching with packet fragmentation and reassembly.
14. Program to input and validate IPv4
15. Program to input and validate IPv6 addresses

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Computer Fundamentals and Programming in C** | **Course Code** | **24CSC201DS03** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**
The objective of this course is to provide students with a foundational understanding of computer concepts and programming in C. Students will learn about computer hardware and software, operating systems, and various programming languages. The course will also cover problem-solving techniques, C programming fundamentals, and advanced topics such as arrays, functions, pointers, structures, and file handling.

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

CO1: Understand the basic concepts of data and information, the components and classification of computers.
CO2: Develop problem-solving skills using flowcharts, decision tables, pseudocode, and algorithms.
CO3: Demonstrate proficiency in C programming using arrays, strings, pointers and library functions.
CO4: Perform file operations using C library functions to construct robust programs.
CO5: Develop efficient C programs for newer problems/tasks.

## Unit-I

**Computer Fundamentals:** Concept of data and information; Components of Computer: Hardware Input Device, Output Device. CPU: Components of CPU; Memory and Storage Devices; Computer Software: System Software and Application Software; Functions of Operating System. Programming Languages: Machine, Assembly, High Level Language, 4GL; Language Translator; Linker, Loader;

**Classification of Computers**: Micro, Mini, Mainframe, Super computer. Advantages of Computer, Limitations of Computer, Range of Applications of Computer, Social concerns of Computer Technology: Positive and Negative Impacts, Computer Crimes, Viruses and their remedial solutions.

## Unit-II

**Problem Solving:** Problem Identification, Analysis, Flowcharts, Decision Tables, Pseudo codes and algorithms, Program Coding, Program Testing and Execution.

**C Programming Fundamentals:** Keywords, Variables and Constants, Structure of a C program. Operators & Expressions: Arithmetic, Unary, Logical, Bit-wise, Assignment & Conditional Operators, Library Functions, Control Statements: Looping using while, do…while, for statements, Nested loops; decision making using if…else, Else If Ladder; Switch, break, Continue and Goto statements.

## Unit-III

**Arrays & Functions:** Declaration and Initialization; Multidimensional Arrays. String: Operations of Strings;

**Functions:** Defining & Accessing User defined functions, Function Prototype, Passing Arguments, Passing array as argument, Recursion, Use of Library Functions; Macro vs. Functions.

**Pointers:** Declarations, Operations on Pointers, Passing to a function, Pointers & Arrays, Array of Pointers, Array accessing through pointers, Pointer to functions, Function returning pointers, Dynamic Memory Allocations.

## Unit-IV

**Structures and Union:** Defining and Initializing Structure, Array within Structure, Array of Structure, Nesting of Structure, Pointer to Structure, Passing structure and its pointer to Functions; Unions: Introduction to Unions and its Utilities.

**Files Handing:** Opening and closing file in C; Create, Read and Write data to a file; Modes of Files, Operations on file using C Library Functions; Working with Command Line Arguments. Program Debugging and types of errors.

**Suggested Readings:**

1. Gill Nasib Singh: Computing Fundamentals and Programming in C, Khanna Book Publishing Company (Private) Limited, New Delhi.
2. K.A. Lambert, C Problem Solving and Programming, Prentice Hall.
3. B.S. Gottfried, Theory and Problems of Programming with C, Schaum's Outline Series, McGraw-Hill.
4. Gill Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing Company (Pvt.) Limited, New Delhi.
5. Donald Sanders, Computers Today, Tata McGraw-Hill.
6. D. Cooper and H. Mullish, The Spirit of C: An Introduction to Modern Programming, Jaico Publishing House, New Delhi. Gill, Nasib Singh: Essentials of Computer and Network Technology, Khanna Books Publishing Co.(P), New Delhi.
7. Gill, Nasib Singh: Essentials of Computer and Network Technology, Khanna Books Publishing Co.(P), New Delhi.
8. B.W. Kernighan and D.M. Ritchie, The C Programming Language, Prentice Hall India Learning Private Limited.
9. B.S. Gottfried, Theory and Problems of Programming with C, Schaum's Outline Series, McGraw-Hill.
10. E. Balaguruswamy, Programming in C, Tata McGraw-Hill Education.
11. H. Schildt, C: The Complete Reference, Tata McGraw-Hill Education.
12. Y. Kanetkar, Let Us C, BPB Publications.
13. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of programs/ Lab Assignments**

1. Write a C program to demonstrate the use of `printf()` and `scanf()` functions. Prompt the user to enter their name and age, then display the entered information.
2. Write a C program to define and use symbolic constants. Define symbolic constants for PI and the radius of a circle, then calculate and display the area of the circle.
3. Write a C program to demonstrate the use of various operators and expressions. Perform addition, subtraction, multiplication, and division on two numbers entered by the user and display the results.
4. Write a C program to implement decision making using if-else statements. Check if a number entered by the user is positive, negative, or zero and display the result.
5. Write a C program to implement a nested if statement.
6. Write a C program to demonstrate the use of switch statement. Simulate a simple calculator to perform addition, subtraction, multiplication, and division based on user input.
7. Write a C program to implement looping using a while loop.
8. Write a C program to implement looping using a do-while loop.
9. Write a C program to implement looping using for loop
10. Write a C program to demonstrate the use of break and continue statements. Print numbers from 1 to 10, but skip the number 5 using continue and stop the loop when it reaches 8 using break.
11. Write a C program to define and use a function to calculate the factorial of a number. Prompt the user to enter a number and then use a function to calculate and display its factorial.
12. Write a C program to demonstrate the use of arrays. Read 10 integers from the user, store them in an array, and then find and display the largest number.
13. Write a C program to demonstrate string manipulation functions. Read a string from the user and then display its length, and convert it to uppercase and lowercase.
14. Write a C program to demonstrate the use of pointers. Declare an integer variable, assign a value to it, and then use a pointer to display its value and address.
15. Write a C program to pass arrays to a function. Read 5 integers into an array, pass the array to a function, and calculate and display the sum of the elements in the array.

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Computer Organization and Architecture** | **Course Code** | **24CSC201DS04** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25(20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objective:**
The objective of this course is to provide students with a thorough understanding of the fundamental concepts of computer organization and architecture. Students will explore the internal workings of computer systems, including hardware components, instruction sets, memory organization, and input/output mechanisms. Additionally, the course will cover various architectural techniques to enhance performance and efficiency.

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

CO1: Understand the basic structure and operations of a computer system.
CO2: Demonstrate knowledge about various instruction sets, addressing modes and learning to develop Assembly programs.
CO3: Analyze different types of memory organization and its roles in computer performance and efficiency.
CO4: Learn the principles of computer architecture and explore concept of parallel processing.
CO5: Critically analyse various aspects of advanced computer architecture.

## Unit-I

**Representation of Information:** Number Systems: Binary, Octal and Hexadecimal, Integer and Floating-point representation, Character codes: ASCII and EBCDIC.

**Basic Building Blocks and Circuit Design**: Boolean Algebra and Logic Gates: OR, AND, NOT, XOR Gates; De Morgan’s theorem; Universal building blocks; Simplifying logic circuits : sum of product and product of sum form; Karnaugh Map simplification; Combinational logic blocks (Adders, Multiplexers, Encoders, Decoder), Sequential logic blocks (Latches, Flip-Flops, Registers, Counters).

## Unit-II

**Register transfer and Micro-operations:** Register Transfer Language; Bus and memory Transfer; Micro operations: Arithmetic, Logic & Shift Micro operations.

**Basic Computer Organization and Design:** Instructions Codes, Register reference, Memory Reference & Input-Output instructions, Instruction Cycle, Timing and Control, Interrupts; Design of Control unit: Hardwired control unit, Micro-programmed control unit.

**Register Organization:** General Register Organization, Stack Organization, Instruction Formats, Addressing Modes; Data Transfer & Manipulation Instructions.

**Introduction to x86 Assembly Language programming.**

## Unit-III

**Memory Organization:** Memory Hierarchy, Main Memory, Auxiliary Memory, Cache Memory, Virtual Memory.

**Input-Output Organization:** Peripheral Devices, Input-Output interface, Asynchronous Data Transfer, Modes of transfer, Priority interrupt, Direct Memory Access (DMA), input-output processors (IOP), Serial communication.

## Unit-IV

**Parallel Computing:** CISC and RISC - Features and Comparison, Pipeline and Vector Processing: Parallel processing, Pipelining, Arithmetic Pipeline, Instruction pipeline and Arrays Processors.

**Advanced Architecture** Multi-processors, characteristics of multi-processors, Interconnection structures, Inter-processor Arbitration, Inter-processor Communication and Synchronization, Cache Coherence.

**Suggested Readings:**

1. M.M. Mano, Digital Logic and Computer Design, Prentice-Hall of India.
2. Gill, Nasib Singh and Dixit J.B.: Digital Design and Computer Organisation, University Science Press (Laxmi Publications), New Delhi.
3. M.M. Mano, Digital Design, Prentice-Hall of India.
4. A. Kumar, Fundamentals of Digital Circuits, PHI Learning Private Limited.
5. R. Tokheim, Digital Electronics, Tata McGraw-Hill Education. (Assuming TMH stands for Tata McGraw-Hill Education)
6. S. Rangnekar, Digital Electronics, ISTE/ EXCE.L
7. C. Hamacher, Z. Vranesic, and S. Zaky, Computer Organization, McGraw-Hill.
8. W. Stallings, Computer Organization and Architecture - Designing for Performance, Prentice Hall of India.
9. D.A. Patterson and J.L. Hennessy, Computer Organization and Design –The Hardware/Software Interface, Morgan Kaufmann Publishers.
10. J.P. Hayes, Computer Architecture and Organization, McGraw-Hill.
11. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of x86 Assembly programs/Lab Assignments**

1. Write a simple program to display a message on console.
2. Load, store, move, and exchange data between registers and memory.
3. Write a program to display a pyramid of numerals or alphabets of a given string
4. Learn Basic syntax, instruction formats, and simple programs in Assembly Language of x86 Machines
5. Write a program using Macros.
6. Write a program to Load, store, move, and exchange data between registers and memory.
7. Write a program to perform addition, subtraction, multiplication, and division Using x86 Arithmetic Instructions
8. Write a program to AND, OR, XOR, NOT, shift, and rotate operations using x86 Logical and Bit Manipulation Instructions
9. Implementing Branching in x86 Assembly Language
10. Implementation of Loop Structures in x86 Assembly Language
11. Perform Array Processing in x86 Assembly Language: Accessing, iterating, and manipulating arrays.
12. Development of Procedures and Macros in x86 Assembly Language
13. Using x86 String Instruction to perform operations on strings using MOVS, LODS, STOS, SCAS, and CMPS instructions.
14. Implementing a Palindrome program for a given input.
15. Implementing a program for searching an element in a given array.

***Any other programs assigned by the teachers.***

| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
|---|---|---|---|
| **Name of the Course** | **Database Management Systems** | **Course Code** | **24CSC201DS05** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objective:**
The objective of this course is to provide students with a comprehensive understanding of database management systems (DBMS). Students will learn about database architecture, data modelling, relational models, schema refinement, transaction management, concurrency control, and database security. Additionally, the course will cover practical aspects of using Oracle DBMS, including SQL queries and form construction.

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

CO1: Understand Database Concepts and Architectures.
CO2: Design and Implement Databases.
CO3: Understand and Implement Database Security and Integrity.
CO4: Master Database Backup and Recovery Techniques.
CO5: Efficiently Query and Manipulate Data, develop and optimize Database applications.

## Unit-I

**Introduction:** Characteristics of database approach, data models, DBMS architecture and data independence. Database Languages, Classification of DBMS, Database Users and Administrator.

**DBMS Environment:** Database Access for applications Programs, Transaction Management, Database system Structure, Storage Manager, Query Processor.

**E-R Modeling:** Entity types, Entity set, attribute and key, Relationships, Relation types, Roles and Structural constraints, Weak entities, Enhanced ER Model.

## Unit-II

**Relational Model:** Introduction to the Relational Model, Integrity Constraint over Relations, Enforcing Integrity constraints, Querying relational data, Introduction to views, Destroying/altering Tables and Views.

**Relational Algebra and Calculus:** Relational Algebra, Set operations, Selection and projection, renaming, Joins, Division, Examples of Algebra overviews, Relational calculus: Tuple relational Calculus, Domain relational calculus, Expressive Power of Algebra and Calculus.

## Unit-III

**Schema Refinement & Normalization:** Problems Caused by redundancy, Schema refinement in Database Design, Decompositions &amp; its properties, Problem related to decomposition, Functional Dependency.

**Normalization:** First, Second, Third Normal forms, BCNF, Lossless join Decomposition, Dependency preserving Decomposition, Multi valued Dependencies, Fourth Normal Form.

**Transaction Management:** ACID Properties, Transactions and Schedules, Concurrent Execution of transaction, Serializability and recoverability.

## Unit-IV

**Concurrency Control:** Introduction to Lock Management, Lock Conversions, Dealing with Dead Locks, Concurrency without Locking, Recovery Techniques, Database Security.

**Introduction to MySQL/Oracle:** Working with MySQL/Oracle. Getting started, Modules of MySQL/Oracle, Invoking SQL*Plus/MySQL Command-line client (‘mysql’), Data types, Data Constraints, Operators, Data manipulation - Create, Modify, Insert, Delete and Update; Searching, Matching and Oracle Functions.

**Suggested Readings:**

1. R. Krishnan, Database Management Systems, by J. Gehrke, Tata McGraw-Hill.
2. A. Silberschatz and H.F. Korth, Database System Concepts, McGraw-Hill.
3. P. Radha Krishna, Database Management Systems, HI-TECH Publications.
4. C.J. Date, Introduction to Database Systems, Pearson Education.
5. P. Rob and C. Coronel, Database Systems: Design, Implementation, and Management, Thomson.
6. R. Elmasri and S.B. Navathe, Database Management Systems, Pearson Education.
7. M. Leon, Database Management Systems, Leon Vikas Publishers.
8. T.M. Connolly, Database Systems, Pearson Education.
9. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of programs (Using MySQL/Oracle)**

1. Write SQL command to create a table with columns of your choice. Write an SQL command to add anew column to the table. Write an SQL command to delete the table from the database.
2. Write SQL command to insert a new record into the table. Write an SQL command to update a column value in the table. Write an SQL command to delete a record from the table.
3. Write SQL commands showcase the use `ROUND`, `COUNT` , `UPPER` , `SYSDATE` , and `TO_CHAR` functions
4. Write SQL command to perform an arithmetic calculation on numeric columns.
5. Write SQL command to find records that satisfy either of two conditions using logical operators.
6. Write SQL command using the `BETWEEN` operator to find records within a specific range.
7. Write SQL command to perform a natural join between two tables.
8. Write SQL command to group records by a column and count the number of records in each group, displaying only groups that meet a specific condition.
9. Write SQL command to order records by a specific column in ascending or descending order.
10. Write SQL command to create a view that displays specific columns from a table.
11. Write SQL commands to implement `PRIMARY KEY`, `FOREIGN KEY`, `UNIQUE`, `CHECK`, and `NOT NULL` constraints on a table.
12. Write SQL commands to demonstrate transaction control using `ROLLBACK`, `COMMIT`, and `SAVEPOINT`.
13. Write SQL command to create a new database or tablespace.
14. Write SQL commands to create a user and delete a user.
15. Write SQL commands to grant and revoke roles.

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Office Automation Tools** | **Course Code** | **24CSC202SE01** |
| **Hours/Week** | **2+0+4** | **Credits (L:T:P)** | **2:0:2** |
| **Max. Marks.** | **Theory: 50 (35+15)**<br>**Practical: 50 (35+15)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objective:**

Office automation course enables students in crafting professional word documents, excel spread sheets, power point presentations using the Microsoft suite of office tools. To familiarize the students in preparation of documents and presentations with office automation tools.

**Course Outcomes:**

By the end of the course the students will be able to:

CO1: Understand the fundamental concepts of window operating system.
CO2: Understand and use of various functions of windows.
CO3. Use Word processing tool for text processing.
CO4: Learn Data analysis tool for effective data analysis.
CO5: Learn about Presentation tool for creating presentations.

### Unit-I

**MS-Windows:** Operating system-Definition & functions, basics of Windows. Basic components of windows, icons, types of icons, taskbar, activating windows, using desktop, title bar, running applications, exploring computer, managing files and folders, copying and moving files and folders. Control panel – display properties, adding and removing software and hardware, setting date and time, screensaver and appearance. Using windows accessories.

### Unit-II

**Using MS-Word:** Introduction to word processing interface, Toolbars Creating & Editing Document, Formatting Document, Finding and replacing text, Format painter, Header and footer, Drop cap, Auto-text, Autocorrect, Spelling and Grammar Tool, Document Dictionary, Page Formatting, Bookmark, Previewing and printing document, Advance Features of MS-Word-Mail Merge, Macros and Tables.

### Unit-III

**Using MS-Excel:** Introduction to MS-Excel, Cell, cell address, Creating & Editing Worksheet, Formatting and Essential Operations, Moving and copying data in excel, Header and footer, Formulas and Functions, Charts, Cell referencing, Page setup, Macros, Advance features of MS-Excel-Pivot table & Pivot Chart, Linking and Consolidation, Database Management using Excel-Sorting, Filtering, Validation, What if analysis with Goal Seek.

### Unit-IV

**Using MS-PowerPoint:** Presentations, Creating, Manipulating & Enhancing Slides, Organizational Charts, Excel Charts, Word Art, Layering art Objects, Animations and Sounds, Inserting Animated Pictures or Accessing through Object, Inserting Recorded Sound Effect or In-Built Sound Effect.

**Suggested Readings:**

1. Microsoft Office – Complete Reference – BPB Publication
2. Russell A. Stultz: Learn Microsoft Office — BPB Publication
3. Courter, G Marquis: Microsoft Office 2000, Professional Edition. BPB.
4. Koers, D: Microsoft Office XP Fast and Easy. PHI.
5. Nelson, S L and Kelly, J: Office XP: The Complete Reference. Tata McGraw- Hill.
6. Any other book covering the contents of the subject.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of programs/Lab Assignments**

1. Create a document in MS-Word introducing the word processing interface.
2. Write a step-by-step guide on how to find and replace text in MS-Word. Include examples and screenshots.
3. Create a document in MS-Word showcasing the use of Format Painter, Header and Footer, Drop Cap, AutoText, and Autocorrect features.
4. Write a tutorial document in MS-Word demonstrating the use of the Spelling and Grammar Tool, Document Dictionary, and Page Formatting options.
5. Create a document in MS-Word with bookmarks and demonstrate how to navigate through them. Include a section on previewing and printing documents.
6. Write a guide on using Mail Merge in MS-Word to create personalized letters. Include steps for merging data from an Excel sheet.
7. Create a document in MS-Word with macros to automate repetitive tasks. Explain how to record and execute macros.
8. Design a table in MS-Word to organize data and demonstrate sorting, filtering, and validation functionalities.
9. Create a budget spreadsheet in MS-Excel, including essential operations like creating & editing worksheets, formatting cells, and formulas.
10. Design a worksheet in MS-Excel with a header and footer. Demonstrate how to customize headers and footers for printing.
11. Create a spreadsheet in MS-Excel with various formulas and functions. Include examples of basic arithmetic operations and statistical functions.
12. Develop a chart in MS-Excel to visualize data. Include bar, line, and pie charts, and explain when to use each type.
13. Create a PowerPoint presentation with slides showcasing different slide manipulation features like transitions, animations, and organizational charts.
14. Design a presentation in MS-PowerPoint with Excel charts and WordArt. Demonstrate how to import and manipulate these objects.
15. Create a PowerPoint presentation with animated pictures and sound effects. Include slides with recorded sound effects and animated pictures accessed through objects.

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Web Development-I** | **Course Code** | **24CSC202MV01** |
| **Hours/Week** | **2+0+4** | **Credits (L:T:P)** | **2:0:2** |
| **Max. Marks.** | **Theory: 50 (35+15)**<br>**Practical: 50 (35+15)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**
The objective of this course is to provide students with a comprehensive understanding of the fundamental concepts of internet technologies and web design. Students will learn about the evolution and workings of the internet, the functionality and configuration of internet connections, web browsing techniques, and the tools and technologies used for online communication. Additionally, the course will cover the basics of HTML, CSS, JavaScript, and XML, enabling students to create, design, and publish professional-quality web pages.

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

CO1: Understand the evolution of the internet, various types of computer networks, and their protocols, and configure and manage internet connections and email services.
CO2: Utilize web browsers and search engines efficiently, understand web security, and employ internet tools for communication such as online chatting, messaging, and video conferencing.
CO3: Develop and publish web pages using HTML, including the use of various HTML elements, linking techniques, and layout design.
CO4: Apply CSS for enhancing web page presentation and implement client-side programming using JavaScript.
CO5: Understand the basics of XML and its relevance in web design.

**Unit-I**

**Introduction:** Internet, Evolution of Internet, Types of Computer Network: LAN, WAN, MAN Internet Protocol, Internet Services, WWW, Working of Internet, Introduction to Intranet, DNS working, Configuring Internet Connection, Connecting LAN to Internet; Client-Server environment: Single User, Multi User, Server, Workstation, Computer Network; Network Topologies; Network Protocols, E-Mail Concepts – Configuring E-Mail Program, Sending and Receiving Files through E-Mail, Fighting Spam, Sorting Mail, E-Mail mailing lists and avoiding E-Mail viruses.

**Unit-II**

**Searching and Web Casting Technique:** Popular web servers, Web Browsers; basic features of browsers: bookmarks, cookies, progress indicators, customization of browsers, browsing tricks, next generation web browsing, search engines; Hypertext Transfer Protocol (HTTP), URL.

**Internet Tools:** Online Chatting, Messaging, and Conferencing Concepts, Usenet newsgroup concepts: Reading UseNet newsgroups, Instant messaging, Web-Based chat rooms and discussion boards, Voice and Video conferencing. Streamlining Browsing, Keeping track of Favourite Websites, Web Security, Privacy, and Site-Blocking.

**Unit-III**

**Web Designing using HTML:** Understanding HTML, XHTML Syntax and Semantics, HTML Elements: Paragraph, Lists, Tables, Images, Frames, Forms, Linking to other Web Pages: External and Internal linking, E-mail Links; Working with Background colors and Images; Marquee; Text Alignment and Text Formatting, Advanced Layout with Tables; Publishing HTML Pages.

**Unit-IV**

**Cascading Style Sheets (CSS):** Introduction, Inline, Internal, External CSS, Linking CSS to Web Page.

**Client–Side Programming:** Introduction to JavaScript, Basic Syntax, Variables and Data types, Statements, Operators, Literals, Functions, Objects, Arrays.

**XML:** Relation between XML and HTML, Goals of XML, Structure and Syntax of XML, Well Formed XML, DTD and its Structure, tree structures in data organization, Searching with XPath.

**Suggested Readings:**

1. R. Greenlaw and E. Hepp, "Fundamentals of the Internet and the World Wide Web," THM.
2. Gill, Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing Company (Pvt.) Limited, New Delhi.
3. P. J. Deitel, H. M. Deitel, and A. B. Nieto, "Internet & World Wide Web Programming," Pearson Education.
4. A. Weiss, "Complete Reference Guide to JavaScript," QUE.
5. D. Oliver, "Tech Yourself HTML 4 in 24 Hours," Techmedia.
6. S. Jain, "O' Level Information Technology," BPB Publications.
7. C. Zacker, "10 Minutes Guide to HTML Style Sheets," PHI Learning Pvt. Ltd.
8. V. K. Jain, "O' Level Information Technology," BPB Publications.
9. Any other book covering the contents of the subject.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of programs/Lab Assignments**

1. Write HTML code to display your education details in a tabular format.
2. Write HTML code to display your CV on a web page.
3. Write HTML code to create a Home page having three links: About Us, Our Services and Contact Us. Create separate web pages for the three links.
4. Write HTML code to create a login form. On submitting the form, the user should get navigated to a profile page.
5. Write HTML code to create a Registration Form. On submitting the form, the user should be asked to login with new credentials.
6. Write HTML code to create your Institute website, Department Website and Tutorial website for specific subject.
7. Write HTML code to illustrate the usage of the following:
   ●Ordered List · Unordered List · Definition List
8. Write HTML code to create a frameset having header, navigation and content sections.
9. Write HTML code to demonstrate the usage of inline CSS.
10. Write HTML code to demonstrate the usage of internal CSS.
11. Write HTML code to demonstrate the usage of external CSS.
12. Write HTML program to create a webpage to show different art forms of India, with appropriate title on the title bar. Use different heading tags for the headings, and list them using ordered list.
13. Write HTML program to create sections in the document using appropriate tags and apply different color as background to them. Use internal hyperlinks to move to different points within the page.
14. Write a JavaScript program to declare variables of different data types and display their values in the console.
15. Write a JavaScript program to perform arithmetic operations and display the results.
16. Write a JavaScript function to calculate the factorial of a number and display the result.
17. Write a JavaScript program to create an array of numbers and find the maximum and minimum values.
18. Create a simple XML document to represent a list of books.
19. Create a DTD for the XML document in Assignment 4 and validate the XML document against the DTD.
20. Create an XML document to represent a company organizational chart with nested employee elements.
21. Create a web page that contains the image of a human. Form an image map such that the user clicks on any part of the body, a web page showing information on that part of the body is displayed.
22. Create a registration form for your institution which will ask first name, last name, age, sex, date of birth, class, address, contact number, email id from the candidate. When you submit the form it should display the confirmation message that “You details are saved” (use JavaScript for displaying the confirmation message).
23. Create an HTML page with a paragraph. Apply a style directly using inline styles to change the color and font size. Then use internal CSS to change the background color of the paragraph.
24. Create a nested HTML structure (e.g., a <div> containing a <p> and an <a>). Apply CSS styles to the parent <div> and observe how styles are inherited by child elements. Modify styles on child elements to see how inheritance works.

25. Create a webpage that includes an iframe displaying another webpage. Ensure the iframe has a set width and height, and add a border around it.
26. Create an HTML page and use different types of CSS selectors (universal, type, class, ID, attribute, descendant, child, sibling). Apply styles using these selectors and Design an HTML table and use CSS to style it. Apply borders, padding, background colors, and hover effects
27. Create an HTML page with a background image and a foreground text. Use CSS to set the opacity of the background image.
28. Design an HTML page with overlapping elements. Use CSS to apply different z-index values to control the stacking order of the elements.
29. Create a HTML form that has number of Textboxes. When the form runs in the Browser fill the textboxes with data. Write JavaScript code that verifies that all textboxes has been filled. If a textboxes has been left empty, popup an alert indicating which textbox has been left empty
30. Create an HTML page and use CSS media queries to apply different styles for different screen sizes (e.g., mobile, tablet, desktop).
31. Create a form having number of elements (Textboxes, Radio buttons, Checkboxes, and so on). Write JavaScript code to count the number of elements in a form
32. Write a JavaScript program that creates an array of 5 numbers. Use array methods to add a number to the end, remove a number from the beginning, and find the index of a particular number.
33. Write a JavaScript function that changes the background color of a div element when a button is clicked. Provide multiple buttons to change to different colors.

***Any other programs assigned by the teachers.***

# Semester: Second
# Session: 2024-2025

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Data Structures Using C** | **Course Code** | **24CSC202DS01** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**
The course aims to empower students with a comprehensive skill set in data structures, fostering both theoretical understanding and practical application, preparing them for analyzing and applying algorithms, design and apply efficient algorithms using data structures, understand the significance of efficient data and file organization, develop coding proficiency in data structure applications.

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

CO1: Understand the fundamental concepts of data structures.
CO2: Design and implement various data structures to solve computational problems related to various sorting and searching techniques.
CO3: Apply data structures for efficient storage and retrieval of information.
CO4: Develop programs using C for implementation of all types of data structures covered.
CO5: Implement file handling operations in a C programming environment.

## Unit-I

**Background and Concept of Data Structures:** Data Structure, Concepts of Data Types, Abstract Data Type and their uses, Background for Data Structure, Definition and use of ADT, Array as an ADT, Structure, Pointer.

**Algorithms:** Introduction to Algorithm and their properties, Concepts of Analysis of algorithm with asymptotic notations (Big Oh) and their properties, time and space complexities.

**Stack:** Definition and Primitive Operations, Stack as an ADT, Stack Applications: Evaluation of Infix, Postfix and Prefix expressions, converting from infix to prefix and postfix.

## Unit-II

**Queue:** Definition, Queue as an ADT and Primitive Operations of Linear and Circular Queue, Application and advantages of Linear, Circular Queue, and Priority Queue (Ascending and Descending Priority Queue).

**Recursion:** Definition and Principle of Recursion, Application of Recursion, Recursion removal using stack, example of recursion for TOH Factorial, Fibonacci Sequences, GCD, efficiency of above recursive algorithms.

**Linked List:** List concepts, Definition and List as ADT, Static and Dynamic List Structure and implementation, Types of linked list, Operations on Linked List, Singly linked list, Circular Linked List, Doubly Linked List, Doubly Circular Linked List, Inserting, traversing and deleting nodes at beginning, end and specified positions in these linked lists, Linked implementation of a stack and queue in singly linked list.

## Unit-III

**Tree:** Definition and basic terminologies of tree, Binary Tree: Introduction, Types of Binary Tree, Level and depth, height balance tree(AVL), Operations in Binary Search Tree (BST): Insertion, Deletion, Searching, Tree Traversal: Pre-order traversal , In-order traversal (sorted list of Nodes), Post-order traversal, Applications of Binary Tree (Huff man tree, expression tree).

**Sorting:** Introduction and types of sorting Algorithm and implementation of Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge Sort Comparison and Efficiency of sorting algorithms.

**Searching:** Introduction Sequential Search, Binary Search and Tree Search Comparison and Efficiency of Searching, Hashing: hash function, hash table and collision resolution techniques.

## Unit-IV

**Graph:** Definition, Representation of Graph, Types of Graph, Graph Traversal: Depth First Search, Breadth First Search Spanning Tree, Prim’s Algorithm, Kruskal’s algorithm and Round Robin Algorithm, Shortest Path Algorithm, Greedy and Dijkstra’s Algorithm.

**Overview of File Structures:** Concept of a file, types of files, File operations - open, read, write, close. External storage devices, Concepts of record, file, database and database system.

**File Organization:** Sequential file organisation – structures and processing, Record structures and access methods. Indexed sequential file organisation – structures and processing, Indexing techniques, B-trees and hashing for indexed files. Direct file organisation. Hashed File Organization - Hash function implementation.

**Implementation of Data Structures using C.**

**Suggested Readings:**

1. K.A. Lambert, C Problem Solving and Programming, Prentice Hall.
2. Gill, Nasib Singh, Computing Fundamentals and Programming in C, Khanna Books Publishing Co., New Delhi.
3. B.S. Gottfried, Theory and Problems of Programming with C, Schaum's Outline Series, McGraw-Hill.
4. B.W. Kernighan and D.M. Ritchie, The C Programming Language, Prentice Hall India Learning Private Limited.
5. E. Horowitz and S. Sahni, Fundamentals of Data Structures, Galgotia Publications Pvt. Ltd.
6. R.S. Salaria, Data Structures & Algorithms, Khanna Book Publishing Co. (P) Ltd.
7. P.S. Deshpande and O.G. Kakde, C & Data Structures, Wiley Dreamtech India Private Limited.
8. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, Third Edition, MIT Press.
9. Any other book covering the contents of the subject.

**Note:** Latest and additional good books may be suggested and added from time to time.

## List of C programs for Implementing Data Structures/Lab Assignments

1. Write a C program to declare and initialize an array, then find and print the largest and smallest elements in the array.
2. Write a C program to demonstrate basic data structure operations such as creating, inserting, deleting, and displaying elements in an array.
3. Write a C program to insert and delete an element at a specified position in a linear array.
4. Write a C program to implement and demonstrate a sequential search on an array.
5. Write a C program to implement and demonstrate a binary search on a sorted array.
6. Write a C program to implement and demonstrate the Bubble sort algorithm.
7. Write a C program to perform operations: push, pop, and display.
8. Write a C program to implement a queue using an array and perform basic queue operations: enqueue, dequeue, and display.
9. Write a C program to implement and demonstrate the Selection sort algorithm.
10. Write a C program to implement and demonstrate the Insertion sort algorithm.
11. Write a C program to implement and demonstrate the Quick sort algorithm.
12. Write a C program to implement and demonstrate the Merge sort algorithm.
13. Write a C program to implement a singly linked list and perform insertion, deletion, and traversal operations.
14. Write a C program to implement a binary tree and perform in-order, pre-order, and post-order traversal using recursion.
15. Write a C program to implement a binary search tree (BST) and perform insertion, deletion, and search operations.

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Operating Systems** | **Course Code** | **24CSC202DS02** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**

The objective of this course is to provide students with a comprehensive understanding of the fundamental concepts and functionalities of operating systems. This understanding will equip them to analyze and manage system resources, processes, memory, and I/O devices effectively. The course will also introduce students to the principles of concurrent programming and distributed file systems, along with providing a closer look at the implementation details of a Unix-like operating system.

**Course Outcomes:**

By the end of the course the students will be able to:

CO1: Understand the elementary concepts of an Operating system.
CO2: Perform CPU scheduling to achieve maximum throughput from the system.
CO3: Manage the memory space more effectively and efficiently by implementing paging, Segmentation.
CO4: Compare the performance of any system in terms of different performance evaluators.
CO5: Work efficiently in Unix/Linux environment.

## Unit-I

**Introduction:** Background of operating system, Operating system as Extended Machine and Resource Manager, History of Operating Systems(First, Second, Third and Fourth generation), Hardware review (Processors, Memory, I/O devices, Buses)..

**Evolution of Operating System:** batch system, multiprogramming, time-sharing, real-time, mainframe operating systems, multiprocessor operating systems, handheld, embedded, smartcard, distributed and personal computer operating systems.

**Operating system Concepts:** Booting Computer, Address Spaces, Files, Client-Server Model, Security.

**Operating system Components:** Process Managements, Memory Managements, I/O managements, Operating system services, System calls, System calls for Process, File and Directory management.

Operating system structures: Monolithic system, Layered system, Micro Kernels, Exo Kernels, Virtual Machines, Storage Structures, I/O structures, Files structures, and system Protections.

## Unit-II

**Processes Management:** Process model, Process creation, Process termination, Process states and transition, Thread model, Thread usage, Implementing thread in user space and Kernel.

**Synchronisation:** Interprocess Communication, Race conditions, , Critical regions, Mutual exclusion with busy waiting, Disabling interrupts, Lock variables, Strict alternation, Peterson's solution, Sleep and wakeup, The producer consumer problem, semaphores, Mutexes, monitors, message passing, classical IPC problems: The dining philosopher problem.

**Scheduling:** Process scheduling and Context Switch, Three level scheduling, Scheduling Algorithms: First Come First Serve, Shortest Job First, Priority, Round Robin, Shortest time Remaining First, Multiple queues.

**Deadlocks:** Introduction, Resources, Deadlock characterization, Deadlock modeling, Methods for handling deadlock, Ostrich algorithm, Deadlock prevention and avoidance, Safe and unsafe states, Banker's algorithm for single resource and multiple resources, Deadlock detection and recovery.

## Unit-III

**Memory management:** Address spaces, Monoprogramming without swapping, Multiprogramming with fixed partitions, Swapping, Memory management with bitmaps and linked list, Overlays, Memory allocations, First fit, Next fir, Best fit, Worst fit, Fragmentations, Virtual memory, Paging, Page tables, Paging hardware, TLB, Page replacements algorithms: Principle of optimality, First in First Out, LRU, LFU, NRU, Second Chance Page replacement, Clock, Working set page replacement, Belady's anamoly, Stack algorithm, Segmentation, and segmentation with paging.

**File systems:** File naming, File structure, File types, File access, File attributes, File operations, Access Methods, Directories and Levels, Directories Operations, Single level, two level and hierarchical directory system, File system mounting and sharing, Protection, Access control, File system layout, File system Implementation, Contiguous allocation, Linked list allocation, Linked list allocation using table in memory, Inodes, File system Examples.

## Unit-IV

**Input Output management:** I/O devices, Devices Controller, Memory Mapped I/O, Direct Memory Access (DMA),Interrupts, I/O software Principles: programmed I/O, Interrupt driven I/O, DMA based I/O, I/O Software Layers, Interrupt handlers, Device drivers, Uniform interface for device drivers, Buffering, Allocating and Releasing dedicated devices.

**Disk management:** Disk structure, RAID, Disk scheduling, First come first served, Shortest seek time first, SCAN, C-SCAN, LOOK, C-LOOK, Error handling and formatting, Stable storage management.

**Unix/Linux Operating Systems:** Overview of Unix/Linux in general and implementation of all above functions in these Operating System(s).

**Suggested Readings:**

1. A.S. Tanenbaum, Modern Operating Systems, Prentice-Hall.
2. A.P. Godbole, Operating Systems, Tata McGraw-Hill Education.
3. J.L. Peterson and A. Silberschatz, Operating System Concepts, Addison-Wesley Publishing Company.
4. H.M. Deitel, An Introduction to Operating Systems, Addison-Wesley Publishing Company.
5. Michael Kerrisk: The Linux Programming Interface: A Linux and UNIX System Programming Handbook, No Starch Press.
6. Richard Blum and Christine Bresnahan: Linux Command Line and Shell Scripting Bible, Wiley.
7. Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, and Dan Mackin: UNIX and Linux System Administration Handbook, Addison-Wesley Professional.
8. Robert Love: Linux Kernel Development, Addison-Wesley Professional.
9. Brian Ward: How Linux Works: What Every Superuser Should Know, No Starch Press.
10. B.W. Kernighan and R. Pike, The UNIX Programming Environment, Prentice Hall.
11. M. Bach, The Design of the UNIX Operating System, Prentice Hall.
12. S. Prata, Advanced UNIX Programming Guide, BPB Publications.
13. S. Das, UNIX Concepts and Applications: Featuring SCO UNIX and LINUX, Tata McGraw-Hill Education.

**Note:** Latest and additional good books may be suggested and added from time to time.

## List of programs/Lab Assignments (Using Unix/Linux)

1. To understand and run basic UNIX/LINUX commands
2. To study of Basic Unix/Linux Commands and various Unix/Linux editors such as vi, ed, ex and EMACS.
3. To study basic file manipulation commands of Unix/Linux commands
4. To write C Programs using the following system calls of Unix/Linux operating system fork, exec, getpid, exit, wait.
5. To write C Programs using the following system calls of Unix/Linux operating system close, stat, opendir, readdir.
6. To write C programs to simulate Unix/Linux commands like cp, ls, grep.
7. Write a Shell program to check the given number is even or odd.
8. Write a Shell program to check the given year is leap year or not.
9. Write a Shell program to find the factorial of a number.
10. Write a C program for implementation of Priority scheduling algorithms.
11. To write a C program for implementation of Round Robin scheduling algorithms.
12. To write a C program for implementation of FCFS and SJF scheduling algorithms.
13. To write a C program for implementation of SJF scheduling algorithms.

14. To write a C-program to implement the producer – consumer problem using semaphores.
15. To write a C program to implement banker’s algorithm for deadlock avoidance.

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Object Oriented Programming using C++** | **Course Code** | **24CSC202DS03** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**

The objective of the course is to equips students with the fundamental concepts and practical skills of object-oriented programming (OOP) using the C++ language. Through exploration of key OOP principles like encapsulation, inheritance, and polymorphism, students will gain the ability to design, develop, and implement robust and maintainable C++ applications.

**Course Outcomes:**

By the end of the course the students will be able to:

CO1: Use the characteristics of an object-oriented programming language in a program.
CO2: Use the basic object-oriented design principles in computer problem solving.
CO3: Apply C++ features to program design and implementation.
CO4: Design and implement programs of Constructor, Destructor, and Inheritance.
CO5: Design and implement Polymorphism, Exception handling, Templates and Working with files.

## Unit-I

**Object Oriented Programming Concepts:** Procedural Language and Object-Oriented approach.

**Characteristics of OOP:** Objects, classes, Encapsulation, Data Abstraction, Inheritance, Polymorphism, Dynamic Binding, Message Passing.

**Structure of C++ program:** Data-types, Variables, Static Variables, Operators in C++, Arrays, Strings, Structure, Functions, Recursion, Control Statements.

## Unit-II

**Classes:** Class, object, Memory Allocation for Objects, memory layout of objects, private, public, protected member functions, static members. Constructors: Features, types, dynamic constructor, Parameterized constructors; destructors.

**Memory management:** Dynamic Memory allocation: new, delete, Object Creation at Run Time; This Pointer.

## Unit-III

**Inheritance:** Derived Class and Base Class, Different types of Inheritance, Overriding member function, Public and Private Inheritance, Ambiguity in Multiple inheritance, Virtual Inheritance, Abstract Class.

**Polymorphism:** Definition, operator overloading, Overloading Unary and Binary Operators, Function overloading, Virtual function, Friend function, Static function.

## Unit-IV

**Exception handling:** Throwing, Catching, Re-throwing an exception, specifying exceptions; processing unexpected exceptions; Exceptions when handling exceptions, resource capture and release.

**Templates:** Introduction; Class templates; Function templates; Overloading of template function, namespaces.

**Introduction to Standard Template Library (STL):** benefits of STL; containers, adapters, iterators, vector, lists.

**Suggested Readings:**

1. H. Schildt, C++ - The Complete Reference, Tata McGraw Hill Publications.
2. B. Swamy, C++, Tata McGraw Hill Publications.
3. E. Balaguruswamy, Object Oriented Programming and C++, TMH.

4. S. Shah & A. Thakker, Programming in C++, ISTE/EXCEL.
5. J. Johnston, C++ Programming Today, PHI.
6. A. Olshevsky, Revolutionary Guide to Object Oriented Programming Using C++, SPD/WROX.
7. R. Rajaram, Object Oriented Programming and C++, New Age International.
8. S. Samanta, Object Oriented Programming with C++ & JAVA, PHI.
9. R. Subburaj, Object-Oriented Programming with C++, VIKAS.
10. S. B. Lippman, J. Lajoie, and B. E. Moo, C++ Primer, Addison-Wesley.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of programs/Lab Assignments**

1. Write a C++ program to take two numbers as input from the user and display their sum.
2. Write a C++ program to demonstrate decision-making constructs like if-else and looping constructs like for and while loops.
3. Write a C++ program to define a class called `Rectangle` with attributes `length` and `width`, and display the area of the rectangle.
4. Write a C++ program to demonstrate the concept of inheritance by creating a base class `Shape` and derived class `Rectangle`. Display the area of the rectangle using inheritance.
5. Write a C++ program to define a class called `Student` with attributes `name` and `roll number`. Use member functions to input and display student details.
6. Write a C++ program to demonstrate the use of constructor and destructor in a class.
7. Write a C++ program to showcase the use of access specifiers (`public`, `private`, `protected`) in a class.
8. Write a C++ program to demonstrate function overloading by defining multiple functions with the same name but different parameters.
9. Write a C++ program to demonstrate dynamic polymorphism using virtual functions.
10. Write a C++ program to demonstrate the working of friend function.
11. Write a C++ program to demonstrate the use of pointers to objects. Define a class `Book` with attributes `title` and `author`, and use pointers to access and display book details.
12. Write a C++ program to handle exceptions using `try-catch` blocks.
13. Write C++ code to implement a simple template function to find the maximum of two numbers. Test the function with different data types.
14. Write a C++ program to perform file input and output operations, including opening, reading, writing, and closing files.
15. Implement error handling during file operations in a C++ program, including handling exceptions and error codes.

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Artificial Intelligence** | **Course Code** | **24CSC202DS04** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**
The objective of this course is to introduce you to the fundamental concepts and applications of Artificial Intelligence (AI). You will explore various problem-solving techniques, knowledge representation methods, expert systems, perception, learning, and planning in AI systems. Additionally, you will delve into neural networks and fuzzy logic, equipping you with a foundational understanding of these key AI subfields.

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

CO1: Grasp core AI concepts and problem-solving techniques.
CO2: Utilize knowledge representations methods in AI systems.
CO3: Comprehend the role and development of expert systems and explore perception, and learning concepts in AI.
CO4: Grasp the fundamental of Neural Network and Fuzzy Logic.
CO5: Learning Prolog and applying for AI programming applications.

## Unit-I

**Introduction:** Definition and applications of Artificial Intelligence, Problem solving: Defining problem as State space search, Production systems, Problem characteristics, Search techniques: Brute force and Heuristic search and their different searching techniques.

**Knowledge representation:** Types of knowledge, Inference rule, Knowledge Representation: Logic based Knowledge representation, Rule based knowledge representation;

## Unit-II

**Knowledge representation:** Non-Monotonic reasoning, Knowledge representation based on probability and uncertainty; Knowledge representation schemes: Formal logic, Inference Engine, Semantic net, Frame, Scripts.

**Expert System:** Definition, Role of Knowledge in expert system, Architecture of Expert system.

## Unit-III

**Expert system development life cycle:** Problem selection, Prototype construction, Formalization, Implementation, Evaluation, Knowledge acquisition and different acquisition techniques, Knowledge engineering, Cognitive behaviour.

**Perception:** Sensing, Speech recognition, Vision, Action.

**Learning, Planning and Understanding:** Learning and its different types, Planning, understanding.

## Unit-IV

**Neural Networks:** Introduction, Comparison of artificial neural networks with biological neural networks, Learning in neural networks, Perceptions, Back propagation networks, application of neural networks.

**Fuzzy logic:** Definition, Difference between Boolean and Fuzzy logic, fuzzy subset, fuzzy membership function, fuzzy expert system, Inference process for fuzzy expert system, fuzzy controller.

**Programming with Prolog.**

**Suggested Readings:**

1. E. Rich and K. Knight, Artificial Intelligence, Tata McGraw-Hill.
2. E. Kumar, Introduction to Artificial Intelligence.
3. D. W. Rolston, Principles of Artificial Intelligence and Expert System Development, McGraw-Hill Book

Company.

4. M. Tanimoto, Introduction to AI using LISP
5. D. A. Patterson, Artificial Intelligence and Expert Systems.
6. E. Balagurusamy, Artificial Intelligence & Technology.
7. H. C. Mishkoff, Understanding Artificial Intelligence, BPB Publications.
8. A. Bharti and A. Chaitenya, Natural Language Processing, PHI Learning Pvt. Ltd.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of Programs**

1. Write simple facts for the statements using PROLOG.
2. Write a Program to Solve the 8 Queens Problem using Backtracking.
3. Write a Program for Graph Traversal using Depth First Search (DFS).
4. Write a Program for Graph Traversal using Breadth First Search (BFS).
5. Write a Program for Knowledge Representation using Semantic Network.
6. Write a Program to Solve the Travelling Salesman Problem (TSP).
7. Write a Program to add two numbers.
8. Write a Program to read address of a person using compound object.
9. Write a Program to show concepts of cut and fail operators.
10. Write a Program to count number of elements in a list.
11. Write a Program to append, replace and delete an integer from the list.
12. Write a Program to demonstrate family relationship.
13. Write a program to categorize animal characteristics.
14. Write a Program to convert centigrade temperature to Fahrenheit and vice-versa and checks if the temperature is below freezing or not.
15. Write a Program to implement Factorial and Fibonacci of a given number using PROLOG as per user input.

***Any other programs assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Software Engineering** | **Course Code** | **24CSC202DS05** |
| **Hours/Week** | **3+0+2** | **Credits (L:T:P)** | **3:0:1** |
| **Max. Marks.** | **Theory: 75 (50+25)**<br>**Practical: 25 (20+5)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**

The objective of this course is to equip students with the fundamental principles and practical skills necessary to develop high-quality software applications in a structured and efficient manner. Students will gain a comprehensive understanding of the software development lifecycle, from requirements analysis and project planning to design, coding, testing, and re-engineering.

**Course Outcomes:**

By the end of the course the students will be able to:

CO1: Apply software engineering principles and methodologies to address the challenges of software development.

CO2: Develop skills for effective software project management, including cost estimation, scheduling, and quality assurance practices.

CO3: Apply software design principles and implement well-structured and maintainable software solutions.

CO4: Develop and utilize various software testing methodologies to identify and rectify defects within the code.

CO5: To understand the concepts of Software Re-Engineering and Reverse Engineering and Configuration management.

## Unit-I

**Introduction to Software Engineering**: Software crisis, Software engineering Approach and Challenges, Software development process models with comparison: Waterfall, Prototype, Time boxing and Spiral Models, RAD Model and Automation through software environments. Quality Standards like ISO 9001, SEI-CMM.

**Requirement Analysis:** Structured Analysis, Behavioral & non-behavioral requirements, Software requirement specification: components & characteristics, Function point metric.

## Unit-II

**Software Project Planning:** Cost estimation, static, Single & multivariate models, COCOMO model, Putnam Resource Allocation Model, Risk management, project scheduling, personnel planning, team structure, Software configuration management, quality assurance, project monitoring, Empirical.

**Software Design:** Fundamentals, problem partitioning & abstraction, design methodology, Function Oriented Design, Cohesion, Coupling & their classification, User Interface Design, Detailed design, Information flow metric.

## Unit-III

**Coding:** Choosing Programming Language, Characteristics of Program, Avoiding Dead Codes, and Program Metrics: Size Estimation; Complexity metric (McCabe’s Cyclometic Complexity), Halsted Theory, Function Point Analysis.

**Software Testing:** Impracticality of Testing all Data and Paths, Levels of testing, Functional vs. Structural testing, Static and Dynamic Testing Tools, Regression testing, Mutation Testing, Stress Testing; Validation Vs. verification.

## Unit-IV

**Software Re-Engineering:** Source Code Translation, Program Restructuring, Data Re-Engineering, Reverse Engineering.

**Software Configuration Management:** Maintaining Product Integrity, Change Management, Version Control, Configuration accounting: Reviews, Walkthrough, Inspection, and Configuration Audits; Reliability Models (JM, GO, MUSA Markov), Limitations of Reliability Models.

**Suggested Readings:**

1. Sommerville, Software Engineering, Addison-Wesley.
2. Gill Nasib Singh, Software Engineering (Software Testing, Reliability and Quality Assurance), Khanna Book Publishing Co.(P) Ltd, N. Delhi.
3. J.A. Hoffer, J.S. Valacich, and K. George, Modern Systems Analysis and Design, Pearson Education.
4. R.S. Pressman, Software Engineering: A Practitioner's Approach, Tata McGraw-Hill.
5. P. Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House.
6. P.C. Jorgensen, Software Testing: A Craftsman's Approach, CRC Press.
7. T. Gilb, Principles of Software Engineering Management, Addison-Wesley.
8. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

### List of Software Engineering Lab Assignments/Case Studies/Problems

The students may be assigned any of the following Software Engineering Lab Case Studies to enable them to develop a comprehensive system for managing real-life systems, such as:

1. Online Bookstore Management System
2.Healthcare Management System
3. E-commerce Platform
4. Library Management System
5. Hotel Reservation System
6. Inventory Management System
7. University Course Registration System
8. Banking System
9. Social Networking Site
10. Travel Booking System
Or **any other Case Study assigned by the teachers.**

**Broad Tasks to be covered:**

Students are expected to carry out the following related tasks (as applicable but not only confined to these):

- *Create an SRS document, design ER diagrams, design the database schema, database, design user-friendly interfaces, create UML diagrams, perform code reviews, develop maintenance plans, perform varying kinds of testing, ensure data security, write test cases, and others required in the scope of the specific Case Studies.*

***Any other Case Study or Software Engineering related tasks/activities assigned by the teachers.***

| | | | |
|---|---|---|---|
| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
| **Name of the Course** | **Multimedia and Animation** | **Course Code** | **24CSC202SE01** |
| **Hours/Week** | **2+0+4** | **Credits (L:T:P)** | **2:0:2** |
| **Max. Marks.** | **Theory: 50 (35+15)**<br>**Practical: 50 (35+15)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**

The objective of this course is to equip students with the foundational knowledge and practical skills necessary to create engaging and interactive multimedia experiences. Through exploration of various multimedia components like text, images, audio, and video along with animation techniques.

**Course Outcomes:**

By the end of the course the students will be able to:

CO1: Utilize the various components of multimedia (text, images, audio, video) and basic animation principles to create effective presentations and applications.

CO2: Integrate hypertext for navigation and storytelling within multimedia projects.

CO3: Learn to apply tools and techniques to add pictures, graphics, sound, and animation to multimedia projects.

CO4: Leverage multimedia authoring tools to develop engaging and interactive multimedia presentations and applications.

CO5: Make proficient use of Multimedia/Animation software for creating good multimedia based application.

## Unit-I

**Introduction to Multimedia:** What is multimedia, Multimedia and Hypermedia, Components of multimedia- textual, images, graphics, animation, audio and video, Linear and Non-Linear Multimedia. Application of Multimedia, Requirement of Multimedia System.

**Fonts and Hypertext:** Usage of text in Multimedia, Families and faces of fonts, outline fonts, bitmap fonts, International character sets and hypertext, Digital font’s techniques.

## Unit-II

**Image fundamentals:** Image formats, Bitmap and Vector, Color Models, Color palettes, 2D Graphics, Image Compression and File Formats: GIF, JPEG, JPEG 2000, PNG, TIFF, EXIF, PS, PDF, Basic Image Processing, Use of image editing software, Photo Retouching, Image resolution, Colour, Raster and Vector Graphics.

**Audio Fundamentals:** Audio quality, formats and devices, Digitization of sound, Sound synthesis, Musical Instrument Digital Interface (MIDI), Compression and transmission of audio on internet, Audio Software, Editing sound.

## Unit-III

**Video Fundamental:** Video basics, Formats, how video works, Types of video signals – component, composite and S-video, Video software, Video Recording, Shooting and editing Video, File formats (JPEG, MPEG).

**Animation:** Introduction and definition of animation, Principles, Types and uses, Methods and Techniques of animation, Basic animation, Text and image animation, Time line construction and management, Masking Motion.

## Unit-IV

**Animation Software:** Shape tweening, morphing, Onion skinning, Animation File Formats, Keyframe animation.

**Developing simple** 2D animations, Tweening and motion along a path, creating a motion based movie, adding sound in an animation.

**Working with Multimedia/Animation Software.**

**Suggested Readings:**

1. T. Vaughan, Multimedia: Making It Work, Tata McGraw-Hill.
2. Z.-N. Li and M. S. Drew, Fundamentals of Multimedia, Pearson Education International.
3. R. Aggarwal & B. B. Tiwari, Multimedia Systems, Excel Publications, New Delhi.
4. Z.-N. Li & M. S. Drew, Fundamentals of Multimedia, Pearson Education.
5. P. Ranjan, Principles of Multimedia, Tata McGraw-Hill.
6. M. Mahalakshmi, Multimedia, Margham Publications.
7. M. K. Pakhira, Computer Graphics, Multimedia and Animation, Prentice Hall India Pvt. Ltd.
8. L. Blazer, Animated Storytelling: Simple Steps for Creating Animation and Motion Graphics.
9. A. Beane, 3D Animation Essentials, John Wiley.
10. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

**List of programs/Lab Assignments**
**(Using Multimedia/Animation Software)**

1. Create a simple multimedia presentation incorporating text, images, and audio.
2. Develop a linear multimedia sequence with images and audio narration.
3. Design a non-linear multimedia application using hyperlinks.
4. Explore different multimedia authoring metaphors and create a small project using one.
5. Use multimedia hardware to capture and integrate video footage into a project.
6. Create a text-based multimedia project showcasing different font families and styles.
7. Design a basic hypertext document with links to different sections.
8. Develop an image gallery using different image formats and explain their differences.
9. Implement basic image processing techniques using an image editing software.
10. Create a vector graphic and a bitmap graphic and compare their characteristics.
11. Record and edit an audio clip using audio editing software.
12. Capture and edit a short video clip using video editing software.
13. Create a basic 2D animation using animation software.
14. Develop a timeline-based animation that includes text and image animations.
15. Create a simple 2D motion based animation.

***Any other programs assigned by the teachers.***

| **Name of the Program** | **M.Sc. Computer Science** | **Program Code** | **-----** |
|---|---|---|---|
| **Name of the Course** | **Web Development-II** | **Course Code** | **24CSC202MV02** |
| **Hours/Week** | **2+0+4** | **Credits (L:T:P)** | **2:0:2** |
| **Max. Marks.** | **Theory: 50 (35+15)**<br>**Practical: 50 (35+15)** | **Time of Examination** | **3 Hours** |

**Note:** The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory. All questions will carry equal marks.

**Course Objectives:**

The objective of this course is to builds upon foundational web design principles and equips students with advanced skills for creating dynamic and interactive web experiences. Through exploration of XML, advanced JavaScript features, AJAX/jQuery, server-side scripting with Node.js and PHP, and database interaction with MongoDB, students will gain the necessary capabilities to design and implement data-driven web applications.

**Course Outcomes:**

By the end of the course the students will be able to:

CO1: Design and implement dynamic and interactive web applications using advanced scripting techniques like AJAX and jQuery.

CO2: Utilize XML for data representation and manipulation, integrating it with web services for information exchange.

CO3: Build web servers using Node.js, leveraging its capabilities for asynchronous programming and data persistence with MongoDB.

CO4: Develop server-side scripting skills using PHP, understanding its core syntax and functionalities to create dynamic web pages.

CO5: Get good exposure of real-time/live application development.

## Unit-I

**XML:** Introduction – Syntax - Document structure - Document Type definitions -Namespaces - XML schemas– Displaying raw XML documents - Displaying XML documents with CSS - XSLT style sheets – XML Processors - Web services.

**ECMA Script:** ECMA Script versions, ES5 Features, ES6 introduction, Var Declarations and Hoisting, let declaration, Constant declaration, function with default parameter values, default parameter expressions, unnamed parameters, the spread operator, arrow functions, sets and maps, Array.find(), Array.findIndex(), template strings, Javascript classes, callbacks.

## Unit-II

**AJAX:** type and working of AJAX, handling Ajax request and response, data formats: XML, JSON; Working with JSON data, Loading HTML with Ajax, Loading XML with Ajax, Loading JSON with Ajax, working with data from other servers.

**JQuery :** JQuery, use, finding elements, JQuery selection, getting element content, updating elements, changing content, inserting elements, adding new content, getting and setting attributes, getting and setting CSS properties, using .each(), events, event object, effects, animating CSS properties, using animation, traversing the DOM, working with forms, JavaScript libraries, JQuery and Ajax.

## Unit-III

**Web Servers:** Introduction, HTTP Transactions, Multitier Application Architecture, Client Side Scripting versus Server-Side Scripting, Accessing Web Servers.

**Server Side Scripting with Node.js:** Getting to know node, node.js changed JavaScript forever, features of node, when to use and not use node, asynchronous callbacks, the NoSql movement, node and MongoDB in the wild, Hello World in Node, package.json, modules.

**Built-in Modules:** FS Module, HTTP Module, Events; Node Package Manager (npm), web server using http, node.js with express, middleware, routing in express.

**Node.js with MongoDB:** basics of MongoDB, MongoDB CRUD Operations, Building a data model with MongoDB and Mongoose, Defining simple mongoose schemas.

### Unit-IV

**PHP Programming:** Introduction to PHP; Basic Knowledge of websites; Introduction of Dynamic Website Scope of PHP ; XAMPP and WAMP Installation PHP Functions ; Creating an Array ; Modifying Array Elements ; Processing Arrays with Loops ; Grouping Form Selections with Arrays ; Using Array Functions ; Functions ; Creating User-Defined Functions PHP Programming Basics ; Syntax of PHP ; Embedding PHP in HTML; Embedding HTML in PHP ; Introduction to PHP Variable ; Understanding Data Types ; Using Operators ; Using Conditional Statements ; If(), else if() and else if condition Statement ; Switch() Statements; Using the while() Loop ; Using the for() Loop.

**Suggested Readings:**

1. R. Greenlaw and E. Hepp, "Fundamentals of the Internet and the World Wide Web," THM.
2. Thomas A.Powell “Web Design: The Complete Reference”,4/e. Tata McGraw-Hill
3. P. J. Deitel, H. M. Deitel, and A. B. Nieto, "Internet & World Wide Web Programming," Pearson Education.
4. A. Weiss, "Complete Reference Guide to JavaScript," QUE.
5. V. K. Jain, "O' Level Information Technology," BPB Publications.
6. Any other book(s) covering the contents of the paper in more depth.

**Note:** Latest and additional good books may be suggested and added from time to time.

### List of programs/Lab Assignments

1. Create a simple XML document with a well-defined structure and display it in a web browser.
2. Define a Document Type Definition (DTD) for an XML document and validate the document against it.
3. Create an XML Schema (XSD) for an XML document and validate the document against it.
4. Display an XML document using CSS for styling.
5. Transform an XML document using XSLT to generate HTML output.
6. Create a simple web service that returns an XML response.
7. Write a JavaScript program using ES6 features such as let, const, arrow functions, and template strings.
8. Create an AJAX request to fetch JSON data from a server and display it on a webpage.
9. Use jQuery to manipulate the DOM: select elements, change their content, and apply CSS styles.
10. Implement an AJAX request to load and display XML data on a webpage.
11. Create a simple web application that uses jQuery to handle form submissions and display the results dynamically.
12. Create a basic animation using jQuery to animate an HTML element's CSS properties.
13. Set up a simple HTTP server using Node.js and handle basic GET and POST requests.
14. Create a web application using Node.js and Express that serves static files and handles routing.
15. Connect a Node.js application to a MongoDB database and perform basic CRUD operations.
16. Set up a local development environment using XAMPP or WAMP and write a simple PHP script to display "Hello, World!"
17. Create a PHP script to process form data and display the submitted information.
18. Write a PHP script to create and manipulate arrays, including adding, removing, and modifying elements.
19. Develop a simple PHP application that uses conditional statements to generate dynamic content.
20. Implement a PHP script that connects to a MySQL database and performs basic CRUD operations.
21. Create an XML document to store information about a collection of books, including elements for title, author, publication year, and genre. Write a simple HTML page that uses JavaScript to load and display the XML data.
22. Create an XML file and an XSLT stylesheet to transform the XML data into an HTML table. Use the transformNode or XSLTProcessor methods to apply the transformation and display the result in a web page.
23. Write an HTML page with a script that loads an XML file containing a list of employees. Parse the XML data with JavaScript and display the employee names and departments in a list on the web page.
24. Develop a form that allows users to submit their contact information (name, email, message). Use

AJAX to send the form data to a server-side script (e.g., a PHP script) that echoes the received data. Display the server's response on the web page.

25. Write an Ajax script to get player details from xml file when user select player name in any game. Also create xml file to store details of player(name, country, wickets and runs)
26. Write an AJAX request to a URL that does not exist. Implement error handling to display a user-friendly message when the request fails.
27. Write a jQuery code to select values from a JSONobject. Also Write a JQuery code to remove all CSS classes from an Applications.
28. Create a PHP script that defines variables of different data types (string, integer, float, boolean, array). Output the values and types of these variables using echo and var_dump.
29. Develop an HTML form to upload a file. Write a PHP script to handle the file upload, save the file on the server, and display a message indicating the file upload status.
30. Create a MySQL database and table to store user information (name, email). Write a PHP script to connect to the database and insert a new user record. Display a success message upon successful insertion.
31. Create a web page for Travel agency using PHP.

***Any other programs assigned by the teachers.***