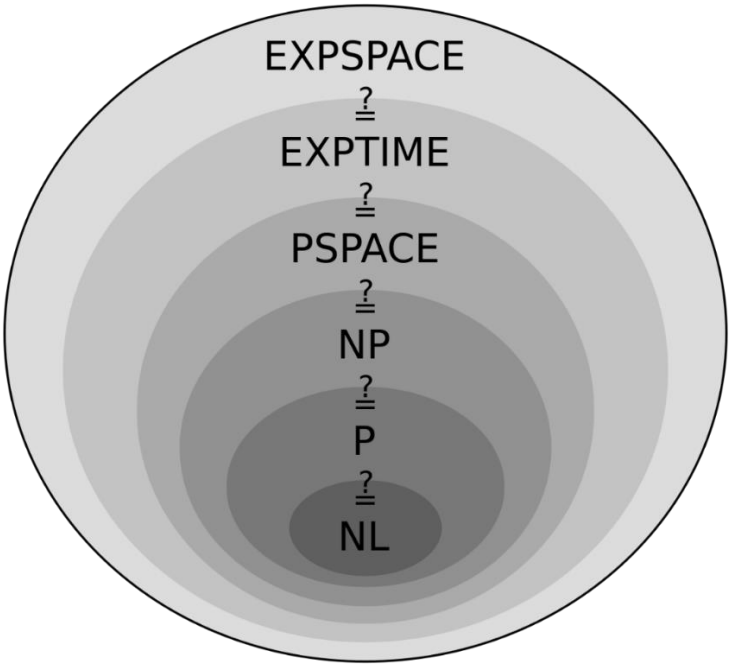
PROGRAMME PROJECT REPORT

Master's in Computer Science

(2 Year Programme in accordance with NEP-2020)



School of Sciences

U. P. Rajarshi Tandon Open University, Prayagraj

Contents

1. Master's Degree Programme

The National Education Policy (NEP) 2020 envisions a new vision that enable an individual to study one or more specialized areas of interest at a deep level and develop capabilities across a range of disciplines including sciences, social sciences, arts, humanities, languages, as well as professional, technical, and vocational subjects. The NEP 2020 focuses on the formulation of expected learning outcomes for all higher education programmes. It states that "National Higher Education Qualifications Framework (NHEQF)" shall be aligned with the National Skills Qualifications Framework (NSQF) to ease the integration of vocational education into higher education. It also points out that higher education qualifications leading to a degree/diploma/certificate shall be described by the NHEQF in terms of Outcome Based Education (OBE).

The design of M.Sc.-Computer Science programme in line with NHEQF offers opportunities and avenues to learn core subjects but also to explore additional avenues of learning beyond the core subjects for holistic development of a learner.

The uniform grading system will also enable potential employers to assess the performance of the learner. In order to bring uniformity in evaluation system and computation of the Cumulative Grade Point Average (CGPA) based on learner's performance in examinations, guidelines framed by the UGC are followed. Hence, adoption of NHEQF helps to overcome the gap between university degree and employability by introducing skills and competencies in the graduates.

2. Master of Science in Computer Science Programme

The structure and duration of postgraduate programme of Master's in Computer Science in accordance with NEP 2020 includes multiple exit options within this period, with appropriate certifications:

- Level 8: a **Bachelor' Degree (Research)** for 4-year programme after completing 4th year of 4-year B.Sc. programme **OR PG Diploma in Computer Science** after completing 1styear (2 semesters) of study of M.Sc. programme.
- Level 9: a **Master of Science in Computer Science** programme after 2 years (4 semesters) of study.

2.1 Programme Mission & Objectives

In line with the mission of the University to provide flexible learning opportunities to all, particularly to those who could not join regular colleges or universities owing to social, economic and other constraints, the 2-year Post-Graduate Programme in Computer Science aims at providing holistic and value-based knowledge and guidance to promote scientific temper in everyday life. The program offers a platform to the learners to fulfill the eligible criteria in various scientific jobs in government and private sector.

The Master of Computer Science programme aims at the following objectives:

- Impart a sound understanding of the core concepts of Computer Science to science, engineering, or other numerate discipline students who have little formal training in computing.
- Inculcate importance of research & development for the welfare of society and understand the contemporary research issues in the different areas of computer science and carry out research in the specialized/emerging areas.

- Work in multidisciplinary and multicultural environment, become entrepreneur based upon societal needs, understanding of professional, social and ethical responsibilities.
- Provide strong core training so that graduates can adapt easily to changes and new demands from industry.
- Equip students with skills to enable them to visualize and to apply new computer technologies to real-world problems through the classroom and experiential learning.

These program objectives acknowledge the interdisciplinarity of computer science and the importance of building a strong foundation with our students.

2.2 Relevance of the Programme with Mission and Goals

The 2-year Post-Graduate Programme in M.Sc.-Computer Science is designed with the objective of equipping learners to cope with the emerging trends and challenges in the scientific domain. In congruence with goals of the University the Programme also focuses to provide skilled manpower to the society to meet global demands. The Programme is designed in such a manner that a successful learner can go for higher studies as well as join the software industry or can run their own start-ups.

2.3 Nature of Prospective Target Group of Learners

The Program is targeted to all individuals looking to earn a postgraduation degree for employment, further higher education, promotion in career, professional development.

2.4 Appropriateness of Programme to be conducted in ODL mode to acquire specific skills & competence

Learning outcomes after Level 8		
Learning Outcomes	**Elements of the descriptor**	**Level 8** **Bachelor' Degree (Research)** **OR** **PG Diploma in Computer Science**
LO 1	Knowledge and understanding	• advanced knowledge about a specialized field of enquiry, with depth in one or more fields of learning within a broad multidisciplinary/interdisciplinary context. • a coherent understanding of the established methods and techniques of research and enquiry applicable to the chosen fields of learning.
LO 2	Skills required to perform and accomplish tasks	• a range of cognitive and technical skills required for performing and accomplishing complex tasks relating to the chosen fields of learning, • cognitive and technical skills relating to the established research methods and techniques,
LO 3	Application of knowledge and skills	• apply the acquired advanced technical and/or theoretical knowledge and a range of cognitive and practical skills to analyze the quantitative and qualitative data gathered drawing on a wide range of sources for identifying problems and issues relating to the chosen fields of learning, • apply advanced knowledge relating to research methods to carry out research and investigations to formulate evidence-based solutions to complex and unpredictable problems.

LO 4	Generic learning outcomes	• listen carefully, read texts and research papers analytically and present complex information in a clear and concise manner to different groups/audiences, • communicate technical information and explanations, and the findings/results of the research studies relating to specialized fields of learning, • present in a concise manner one's views on the relevance and applications of the findings of research and evaluation studies in the context of emerging developments and issues. • pursue self-paced and self- directed learning to upgrade knowledge and skills that will help accomplish complex tasks and pursue higher level of education and research. • problematize, synthesize and articulate issues and design research proposals, • define problems, formulate appropriate and relevant research questions,
LO 5	Constitutional, humanistic, ethical and moral values	• embrace and practice constitutional, humanistic, ethical, and moral values in one's life. • adopt objective, unbiased, and truthful actions in all aspects of work related to the chosen field(s) of learning and professional practice.
LO 6	Employment ready skills, and entrepreneurship skills and mindset	• managing complex technical or professional activities or projects, requiring the exercise of full personal responsibility for output of own work as well as for the outputs of the group as a member of the group/team. • exercising supervision in the context of work having unpredictable changes.

Learning outcomes after Level 9		
Learning Outcomes	**Elements of the descriptor**	**Level 9 (Master's in Computer Science)**
LO 1	Knowledge and understanding	• advanced knowledge about a specialized field of enquiry with a critical understanding of the emerging developments and issues relating to one or more fields of learning, • advanced knowledge and understanding of the research principles, methods, and techniques applicable to the chosen fields of learning or professional practice, • procedural knowledge required for performing and accomplishing complex and specialized professional tasks relating to teaching, and research and development.
LO 2	Skills required to perform and accomplish tasks	• advanced cognitive and technical skills required for performing and accomplishing complex tasks related to the chosen fields of learning, • advanced cognitive and technical skills required for evaluating research findings and designing and conducting relevant research that contributes to the generation of new knowledge, • specialized cognitive and technical skills relating to a body of knowledge and practice to analyse and synthesize complex information and problems.

LO 3	Application of knowledge and skills	• apply the acquired advanced theoretical and/or technical knowledge about a specialized field of enquiry or professional practice and a range of cognitive and practical skills to identify and analyse problems and issues, including real-life problems, associated with the chosen fields of learning.
LO 4	Generic learning outcomes	• listen carefully, read texts and research papers analytically and present complex information in a clear and concise manner to different groups/audiences, • communicate, in a well-structured manner, technical information and explanations, and the findings/ results of the research studies undertaken in the chosen field of study, • meet one's own learning needs relating to the chosen fields of learning, work/vocation, and an area of professional practice, • pursue self-paced and self- directed learning to upgrade knowledge and skills, including research-related skills, required to pursue higher level of education and research.
LO 5	Constitutional, humanistic, ethical and moral values	• embrace and practice constitutional, humanistic, ethical and moral values in one's life, • adopt objective and unbiased actions in all aspects of work related to the chosen fields/subfields of study and professional practice, • participate in actions to address environmental protection and sustainable development issues,
LO 6	Employment ready skills, and entrepreneurship skills and mindset	• adapting to the future of work and responding to the demands of the fast pace of technological developments and innovations that drive shift in employers' demands for skills, particularly with respect to transition towards more technology-assisted work involving the creation of new forms of work and rapidly changing work and production processes. • exercising full personal responsibility for output of own work as well as for group/ team outputs and for managing work that are complex and unpredictable requiring new strategic approaches.

2.5 Instructional Design

2.5.1 2-year M.Sc.-Computer Science Programme Structure

The University follows the credit system in all its programmes. One credit is equal to 30 hours of learner's study time which is equivalent to 15 lectures in conventional system. To earn a master's degree, a learner must earn 80 credits in a minimum of four semesters (two years) with 20 credits per semester. For earning 80 credits, a learner must go through the following Programme Structure:

Programme Structure of M.Sc.-Computer Science under NHEQF

Level	Year	Sem	Core Course 1	Core Course 2	Core Course 3	Research component	Practical Lab/ Dissertation with viva voce	Total credit
8	1	1st	4	4	4	4	4	**20**
		2nd	4	4	4	4	4	**20**
9	2	3rd	4	4	4	4	4	**20**
		4th	4	4	4	4	4	**20**
Total credit								**80**

Explanation of terms used for categorization of courses:

A. **Course 1 to 3:** A course, which should compulsorily be studied by a learner as a core requirement is termed as a Core course.
B. **Research Component:** The components included in this category are Basics in Research (PGBR-01), Mini Project (PGMP-02), Basic Research Tools (PGRT-03).
C. **Practical Lab:** Lab based on theory courses for implementing the algorithms discussed in theory papers.
D. **Industrial Training/ Survey/ Research Project/ Field Work/Apprenticeship/ Dissertation/Internship:** An elective course designed to acquire special/advanced knowledge, such as supplement study/support study to a project work, and a learner studies such a course on his own with an advisory support by a counsellor/faculty member. Currently, Dissertation is offered under code; **MCS121D**.

2.5.2 Course curriculum: The detail of syllabus is given in Appendix-I

2.5.3 Language of Instruction: English. However, learner can write assignment and give Term End Examination (TEE) either in Hindi or English.

2.5.4 Duration of the Programme

Minimum duration in years: 02 Maximum duration in years: 04

2.5.5 Faculty & Support Staff

Professor (1), Assistant Professor (4), and support staff (3)

2.6 Instructional Delivery Mechanisms

The Open University system is more learner-oriented, and the student is an active participant in the teaching-learning process. Most of the instructions are imparted through distance rather than face-to-face communication.

The University follows a multi-media approach to instruction. It comprises of:

- self-instructional printed material (Self Learning Material)
- audio and video lectures
- face-to-face counselling
- assignments
- laboratory work
- Project work in some courses
- teleconference/web conference
- Web Enabled Academic Support Portal
- e-GYANSANGAM (Open Educational Repository): gyansangam.uprtou.ac.in
- e-GYANARJAN: It is a Learning Management System based on Moodle (gyanarjan.uprtou.ac.in) to aid the learner through web conferencing, sharing of learning resources, counselling classes etc.

2.6.1 Self-Learning Material

The Self Learning Material (SLMs) are prepared in line with the UGC guidelines on preparation of SLMs. The prepared study materials are self-instructional in nature.

The course material is divided into blocks. Each block contains a few units. Lessons, which are called Units, are structured to facilitate self-study. The units of a block have similar nature of

contents. The first page of each block indicates the numbers and titles of the units comprising the block. In the first block of each course, we start with course introduction. This is followed by a brief introduction to the block. After the block introduction, emphasis is given on contribution of ancient Indian knowledge into that specific course. Next, each unit begins with an introduction to talk about the contents of the unit. The list of objectives is outlined to expect the learning-based outcome after working through the unit. This is followed by the main body of the unit, which is divided into various sections and sub-sections. Each unit is summarized with the main highlights of the contents.

Each unit has several "Check Your Progress" Questions and Terminal Questions /exercises. These questions help the learner to assess his/her understanding of the subject contents. At the end of units, additional references/books/suggested online web link for MOOCs/Open Educational Resources for additional reading are suggested.

2.6.2 Audio and Video lectures

Apart from SLM, audio and video lectures have been prepared for some courses. The audio-video material is supplementary to print material. The video lectures are available at YouTube channel of university (https://www.youtube.com/@uprtouonlinestudy5413)

2.6.3 Counselling Classes

The face to face (F2F) counselling classes are conducted at head quarter and study centers. The purpose of such a contact class is to answer some of questions and clarify the doubts of learner which may not be possible through any other means of communication. Well experienced counsellors at study centers provide counselling and guidance to the learner in the courses that (s)he has chosen for study. The counselling sessions for each of the courses will be held at suitable intervals throughout the whole academic session. The timetable for counselling classes is displayed at head quarter as well as by the coordinator of study center, however, attending counselling sessions is not compulsory. It is noted that to attend the counselling sessions, the learner must go through the course materials and note down the points to be discussed as it is not a regular class or lecture.

2.6.4 Assignments

The purpose of assignments is to test the comprehension of the learning material that learner receives and help to get through the courses by providing self-feedback to the learner. The course content given in the SLM will be sufficient for answering the assignments.

Assignments constitute the continuous evaluation component of a course. The assignments are available at the SLM section of the home page of the university website. In any case, the learner must submit assignments before appearing in the examination for any course. The assignments of a course carry 30% weightage while 70% weightage is given to the term-end examination (TEE). The marks obtained by the learner in the assignments will be counted in the result. Therefore, it is advised to take assignments seriously. However, there will be no written assignments for Lab courses.

2.6.5 Laboratory Work

Laboratory courses are an integral component of the M.Sc. programme. While designing the curricula for laboratory courses, particular care has been taken to weed out experiments not significant to the present-day state of the discipline. Importance has been given to the utility of an experiment with respect to real life experience, development of experimental skills, and industrial applications. It is planned to phase the laboratory courses during suitable periods

(such as summer or autumn vacations) so that in-service persons can take them without difficulty. Laboratory courses worth 2 credits will require full-time presence of the student at the Study Centre for one week continuously. During this time a student has to work for around 60 hours. Around 40 hours would be spent on experimental work and the remaining time will be used for doing calculations, preparation of records, viewing or listening to the video/audio programmes.

2.6.6 Teleconference/Web conference

Teleconference/web conference, using done through ZOOM/webex in form of online special counselling sessions is another medium to impart instruction to and facilitate learning for a distance learner. The students concerned would be informed about the teleconferencing schedule and the place where it is to be conducted by sending bulk SMS.

2.6.7 Web Enabled Academic Support Portal

The University also provides Web Enabled Academic Support Portal to access the course materials, assignments, and other learning resources.

2.6.8 e-GYANSANGAM

The e-GYANSANGAM (UPRTOU-OER REPOSITORY) is an open access platform for educational resources that rely on the concept of 5Rs namely; Reuse, Revise, Remix, Retain and Redistribute. Uttar Pradesh Rajarshi Tandon Open University in support with Commonwealth Educational Media Centre for Asia initiated the implementation of philosophy behind the NEP-2020 to provide equitable use of technology to support learners (SDG4). This not only ensures inclusive and equitable quality education opportunities but also provides faculty to repurpose high quality open educational resources (OER) such that innovative, interactive and collaborative learning environment is built. UPRTOU believes the philosophy of Antyoday (reaching to last person of the society) and facilitate the learner by providing Self Learning Materials, Lecture Notes, Audio/video Lectures, Assignments, Course materials etc. through face-to-face mode as well as distance mode. This e-GYANSANGAM depository will fulfill the educational facilities through equitable use of technology to the learners.

Objectives

- To provide low-cost access model for learners. To foster the policy of reaching to unreached.
- To break down barriers of affordability and accessibility of educational resources.
- To give faculty the ability to customize course materials for learners.
- To provide equal access to affordable technical, vocational and higher education resources (SDG 4.3).
- To provide ubiquitous access to anyone. This will facilitate the quick availability of educational resources and reduce time.
- To supplement Self Learning Material (SLM).
- To reduce the mentor-mentee gap as depository provide access to a number of local access as well as global access to educational resources.

2.6.9 e-GYANARJAN: It's a Learning Management System based on Moodle (gyanarjan.uprtou.ac.in) to aid the learner through web conferencing, sharing of learning resources, counselling classes etc.

2.6.10 Learner Support Service Systems

(a) Study Centre

A Study Centre has following major functions:

(i) **Counselling:** Counselling is an important aspect of Open University System. Face to face contact-cum-counselling classes for the courses will be provided at the Study Centre. The detailed programme of the contact-cum-counselling sessions will be sent to the learner by the Coordinator of the Study Centre. In these sessions the learner will get an opportunity to discuss with the Counsellors his/her problems pertaining to the courses of study.

(ii) **Evaluation of Assignments:** The evaluation of Tutor Marked Assignments (TMA) will be done by the Counsellors at the Study Centre. The evaluated assignments will be returned to the learner by the Coordinator of Study Centre with tutor comments and marks obtained in TMAs. These comments will help the learner in his/her studies.

(iii) **Library:** Every Study Centre will have a library having relevant course materials, reference books suggested for supplementary reading prepared for the course(s).

(iv) **Information and Advice:** The learner will be given relevant information about the courses offered by the University. Facilities are also provided to give him/her guidance in choosing courses.

(v) **Interaction with fellow students:** In the Study Centre learners will have an opportunity to interact with fellow students. This may lead to the formation of self-help groups.

(b) Learner Support Services (LSS)

The University has formed an LSS cell at the head quarter. The LSS cell coordinates with the Study Centre to get rid of any problem faced by the learner.

2.7 Procedure for admissions, curriculum transaction and evaluation

2.7.1 Admission Procedure

(a) Detailed information regarding admission will be given on the UPRTOU website and on the admission portal. Learners seeking admission shall apply online.

(b) Direct admission to 2-year M.Sc. (Computer Science) program is offered to the interested candidates.

(c) Eligibility: A candidate fulfills either criteria A or B described below:

A. Bachelor of Computer Applications / Bachelor of Engineering/ Bachelor of Technology.

OR

B. Bachelor of Science in Computer Science/Information Technology/ Statistics/Mathematics.

2.7.2 Programme Fee: Rs. 14000/- year. The fee is deposited through online admission portal only.

2.7.3 Evaluation

The evaluation consists of two components: (1) continuous evaluation through assignments, and (2) term-end examination. The learner must pass both in continuous evaluation as well as in the term-end examination of a course to earn the credits assigned to that course. For each course there shall be one written Terminal Examination. The evaluation of every course shall be in two parts, that is 30% internal weightage through assignments and 70% external weightage through terminal exams.

(a) Theory course	**Max. Marks**
Terminal Examination	70
Assignment	30
Total	**100**

(b) Practical course:	**Max. Marks**
Terminal Practical Examination	100

Marks of Terminal Practical Examination shall be awarded as per following scheme:

i.	Write up /theory work	30
ii.	Viva-voce	30
iii.	Execution/Performance/Demonstration	20
iv.	Lab Record	20

The following 10-Point Grading System for evaluating learners' achievement is used for CBCS programmes:

10-Point Grading System in the light of UGC-CBCS Guidelines

Letter Grade	Grade Point	% Range
O (Outstanding)	10	91-100
A+ (Excellent)	9	81-90
A (Very Good)	8	71-80
B+ (Good)	7	61-70
B (Above Average)	6	51-60
C (Average)	5	41-50
P (Pass)	4	36-40
NC (Not Completed)	0	0-35
Ab (Absent)	0	
Q	Qualified	Applicable only for Non-Credit courses
NQ	Not Qualified	

Learner is required to score at least a 'P' grade (36% marks) in both the continuous evaluation (assignments) as well as the term-end examination. In the overall computation also, learner must get at least a 'P' grade in each course to be eligible for the M. Sc. degree.

Computation of CGPA and SGPA

(a) Following formula shall be used for calculation of CGPA and SGPA

For jth semester SGPA (Sj) = Σ (Ci *Gi)/ Σ Ci	where, Ci = number of credits of the ith course in jth semester Gi= grade point scored by the learner in the ith course in jth semester.
CGPA = Σ (Cj *Sj)/ Σ Cj	where, Sj = SGPA of the jth semester Cj = total number of credits in the jth semester

The CGPA and CGPA shall be rounded off up to the two decimal points. (For e.g., if a learner obtained 7.2345, then it will be written as 7.23 or if s(he) obtained 7.23675 then it be will written as 7.24)

CGPA will be converted into percentage according to the following formula:

Equivalent Percentage = CGPA * 9.5

(b) Award of Division

The learner will be awarded division according to the following table:

Division	Classification
1st Division	6.31 or more and less than 10 CGPA
2nd Division	4.73 or more and less than 6.31 CGPA
3rd Division	3.78 or more and less than 4.73 CGPA

2.7.4 Multiple Entry and Multiple Exit options

The 2-year M.Sc. programme is an Outcome-Based Education (OBE) for qualifications of different types. The qualification types and examples of title/nomenclature for qualifications within each type are indicated in Table 1.

Table 1

Level	Qualification title	Programme duration	Entry Option	Exit option
8	B.Sc. (Research) OR PG Diploma in Computer Science	First year (first two semesters) of the M.Sc. Computer Science programme	BCA/ BE/ B.Tech. OR BSc. CS/IT/ Statistics/Mathematics.	Awarded with Bachelor' Degree (Research) of 4 year **OR** Awarded with PG Diploma in Computer Science
9	Master's in Computer Science	Two years (four semesters) of the M.Sc. Computer Science programme	B.Sc. (Research) OR PG Diploma in Computer Science obtained after completing the first year (two semesters) of the M.Sc. programme	Awarded with Master's in Computer Science

2.8 Requirement of the laboratory support and Library Resources

The practical sessions are held in the science laboratories of the Study Centre. In these labs, the learner will have the facility to use the equipment and consumables relevant to the syllabus. The SLM, supplementary text audio and video material of the various courses of the program is available through the online study portal of the University. The University also has a subscription of National Digital Library to provide the learners with the ability to enhance access to information and knowledge of various courses of the programme.

2.9 Cost estimate of the programme and the provisions

2-year M.Sc. programme consists of 16 theory courses, 04laboratory courses and 01 dissertation with vice-viva. Each course is of 4 credits which consists of approx. 12 units. The total approximated expenditure on the development of 16 courses is:

S. No.	Item	Cost per Unit (writing & editing)	Total cost (Rs.)
1	Total no. of units in 16 courses = 192	6500	1248000
2	BOS Meetings etc.	100000	100000
		Total	1348000

2.10 Quality assurance mechanism and expected programme outcomes

(a) **Quality assurance mechanism:** The program structure is developed under the guidance of the Board of studies comprising external expert members of the concerned subjects followed by the School board. The program structure and syllabus is approved by the Academic Council of the University. The course structure and syllabus is reviewed time to time according to the feedback received from the stakeholders and societal needs.

The Centre for Internal Quality Assurance will monitor, improve and enhance effectiveness of the program through the following:

- ✓ Annual academic audit
- ✓ Feedback analysis for quality improvement
- ✓ Regular faculty development programs
- ✓ Standardization of learning resources
- ✓ Periodic revision of program depending upon the changing trends by communicating to the concerned school

(b) **Expected programme outcomes (POs)**

PO 1	Analyze, design and develop new computer technologies to real-world problems.
PO 2	Work in multidisciplinary and multicultural environment or become an entrepreneur based upon societal needs.
PO 3	Develop programming, analytical and logical thinking abilities to learn new technology.
PO 4	Pursue careers in IT industry/ consultancy/ research and development, teaching and allied areas related to computer science.

Programme:	**Master of Science [M.Sc.-Computer Science]**
Year	First Introduction year: 2002
Revision of Programme in accordance with NEP-2020	
Initiation year of revision	2022
Completion year of revision	2023

APPENDIX-I

Academic Year 2023-24

Detailed Programme Structure & Syllabus

Year wise Structure of M.Sc. in Computer Science

Year	Semester	Course Code		Paper Title	Type of Course	Max. Marks	Credits
First	1	MCS-101N		Discrete Mathematics	Theory	100	4
		MCS-102N		C++ and Object-oriented programming	Theory	100	4
		MCS-103N		Data Structures	Theory	100	4
		MCS-104P		Practical Work (Based on 102 & 103)	Practical	100	4
		PGBR-01		Basics in Research	Research	100	4
	2	MCS-106N		Computer Organization	Theory	100	4
		MCS-108N		Data Communication and Computer Networks	Theory	100	4
		MCS -109N		Database Management System	Theory	100	4
		MCS -110P		Practical Work (Based on 109)	Practical	100	4
		PGMP-02		Mini Project	Mini Project	100	4
Second	3	MCS-111N		Design and Analysis of Algorithm	Theory	100	4
		MCS-112N		Java Programming	Theory	100	4
		MCS-113N		Operating System	Theory	100	4
		MCS-115P		Practical Work (Based on 111 & 112)	Practical	100	4
		PGRT-03		Basic Research Tools	Research	100	4
	4	**Compulsory Core Paper**					
		MCS-117N		Soft Computing	Theory	100	4
		MCS-121D		Dissertation with viva voce	Research	100	4
		Select any one group (GROUP A OR GROUP B)					
		Group A	MCS-116N	Computer Graphics	Theory	100	4
			MCS-114N	Multimedia Technology	Theory	100	4
			MCS-119N	Information and Network Security	Theory	100	4
		OR					
		Group B	MCS-104N	Software Engineering	Theory	100	4
			MCS-107N	Theory of Computation	Theory	100	4
			MCS-120N	System Software	Theory	100	4
Total Credit/Max. Marks						**200**	**80**

Syllabus for M.Sc. in Computer Science

Programme: **Master of Science**	Year: **First**	Semester: **I**
Subject:Computer Science		
Course Code: **MCS-101N**	Course Title: **Discrete Mathematics**	
Course Objectives: This course provide students understand discrete objects such as proofs, sets, graphs, colorings, algebraic structures and algorithms that arise naturally and frequently in many areas of mathematics and computer science. It develops a sound understanding of these discrete objects to solve problems arising in computer science.		
Course Outcomes: **CO1** Apply mathematical logic to solve problems. **CO2** Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, and functions. **CO2** Understand and apply counting techniques to the representation and characterization of relational concepts. **CO2** Impart foundations of probabilistic theory which is mostly used in varied applications in engineering and science.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Language of Mathematics and its application**
Unit 1	**Mathematical Logic:** statements, operations, truth values, tautology and quantifiers.
Unit 2	**Arguments:** Rule of Detachment, Validity of a compound statement by using Truth Table, Validity using Simplification Methods, Validity using Rules of Inference, Invalidity of an Argument, Indirect Method of proof and Proof by Counter-Example.
Unit 3	**Boolean Algebra:** Boolean Algbera, Principle of Duality, Isomorphic Boolean Algebras, Boolean Algebra as Lattices, Boolean Functions, Disjunctive Normal Form**,** Conjunctive Normal Form**,** Minimization of Boolean Functions (Karnaugh Map)
Unit 4	**Switching circuits and logical Circuits:** Switching Circuits, Simplification of circuit, Non-Series Parallel Circuits, Relay Circuits, Logic Circuits
Block 2	**Set theory and its application**
Unit 5	**Set theory:** sets, Subsets, Operations on Sets, Complementation, Intersection and Union, Laws Relating Operations, Distributive Laws and De Morgan's Laws.
Unit 6	**Relation:** Relation, binary relations in a Set, Domain and Range of a Relation, Total number of Distinct Relations, Relations as Sets of Ordered Pairs, Types of Relations, Composition of Relations, Equivalence relation in a set, Partition of a Set, Equivalence Class and Quotient set of a set.
Unit 7	**Partitions and Distributions:** Equivalence Relations, Equivalence Classes, Properties of Equivalence Classes, Quotient set and Partition.
Unit 8	**Function:** Functions, Direct and Inverse image, Inverse Functions, Operations on Functions, Composite of functions, Types of Functions and Connection between Equivalence relation and mapping.
Block 3	**Counting Process**
Unit 9	**Mathematical Induction:** Principle of Mathematical Induction, Second Principle of Induction and Well ordering property.
Unit 10	**Combinatorics:** Basic counting principles**,** Principle of Disjunctive counting, Principle of Sequential counting and Ordered and Unordered Partitions.
Unit 11	**Permutation**

Unit 12	**Combination**
Block 4	**Block – 04: Probability theory and application**
Unit 13	**Binomial theorem:** Binomial theorem, General term in a binomial expansion,Middle term in a binomial expansion and Binomial expansion for rational exponents.
Unit 14	**Probability:** Definition of Probability, Addition law for counting and Product law for counting.
Unit 15	**General Counting methods:** General Counting method is the extension part of counting process. It discusses Sum and Product Rules and the Pigeonhole Principle**.**
Unit 16	**The Inclusion- Exclusion Principle:** inclusion-exclusion principle, Alternative form of the inclusion-exclusion principle and Onto Functions.

Suggested Readings:

1. C.L.Liu and D.P.Mohapatra, " Elements of Discrete Mathematics: A Computer Oriented Approach", Mcgraw Hill, Third Edition, 2012.
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications" Mcgraw Hill, Seventh Edition, 2012 (Indian Adaptation by Kamala Krithivasan, Iit Madras).

Suggested online courses (MOOCs)

1. NOC:Discrete Mathematics, IIT Ropar, Prof. Prabuchandran K.J, Prof. Sudarshan Iyengar; https://nptel.ac.in/courses/106106183
2. NOC:Discrete Mathematics, IIT Guwahati, Prof. Benny George K, Prof. Sajith Gopalan https://nptel.ac.in/courses/106103205

This course can be opted as an elective by the students of following subjects: **B.Sc. in Computer Science, B.Sc. in Physics, B.Sc. in Statistics, BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **Master of Science**	Year: **First**	Semester: **I**
Subject: Computer Science		
Course Code: **MCS-102N**	Course Title: **C++ and Object-oriented programming**	
Course Objectives: This course aims to offer a practical mastery of object-oriented concepts such as classes, objects, data abstraction, methods, method overloading, inheritance and polymorphism.		
Course Outcomes: **CO1** Develops a sound approach to problem solving using a middle level programming language. **CO2** Apply techniques like recursion and iteration are learnt to solve a problem. **CO3** Build programming concepts like pointers, structures.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**BLOCK - 1**
Unit 1	**Principles of object-oriented programming:** Object oricnted programming paradigm, Comparison with procedural programming, Basic concepts of object-oriented programming, benefits of O0P, object-oriented Languages, advantage of C++.
Unit 2	**Object Orient Programming System:** Class, inheritance, abstraction, encapsulation and information hiding, polymorphism, overloading.
Unit 3	**Advanced concept**: Dynamism (Dynamic typing., dynamic binding, late binding, dynamic loading). Structuring programs, reusability, organizing object-oriented project,
Block 2	**BIOCK - 2**
Unit 4	**Overview of C++:** Tokens, keywords, identifiers and constants basic data types, user-defined and derived Data types, type compatibility, reference, variables type Casting, operator precedence, control structures, structure, function.
Unit 5	**Classes and objects:** Class specification, class objects, accessing class members, scope resolution operator, data hiding, empty classes, Pointers within a class, passing objects as arguments, returning objects from functions, friend Functions and friend classes, constant parameters and member functions, structures and Classes, static members.
Unit 6	**Object initialization and cleanup**: Constructors destructor, constructor overloading. order of construction and destruction, Constructors with default arguments, nameless objects, dynamic initialization through Constructors, constructors with dynamic operations, constant objects and constructor, static Data members with constructors and destructors, nested classes.
Block 3	**BLOCK - 3**
Unit 7	**Operator overloading and type conversion:** Defining operator overloading, overloading unary operators, overloading binary operators, overloading binary operators using friends, manipulation of strings using Operators, rules for overloading operators. type conversions.
Unit 8	**Inheritance: extending classes:** Deriving derived classes, single multilevel, multiple, hierarchical, hybrid inheritance, Constructors & destructors in derived classes, constructors invocation and data members Initialization, virtual base classes, abstract classes, delegation.
Block 4	**BLOCK- 4**
Unit 9	**Pointers, virtual functions and polymorphism:** Pointers to objects, this pointer. pointers to derived classes, virtual functions, Implementation of run-time polymorphism, pure virtual functions.
Unit 10	**Working with files:** Classes for file stream operations. opening and closing a file, file pointers and their Manipulations, sequential input and output operations, error handling during file Operations, command line arguments.

Unit 11	**Object Oriented Modeling:** Need of object-oriented Modeling, Simulation of real-life problems using OOP concept: Example, Representation of problem using object and class diagrams at design level.

Suggested Readings:

1. The C++ Programming Language by Bjarne Stroustrup, 2013.
2. Programming: Principles and Practice Using C++ by Bjarne Stroustrup, 2014
3. The C Programming Language (Ansi C Version) by Brian W. Kernighan and Dennis M. Ritchie, 1990.
4. The C Programming Language by Brian W. Kernighan and Dennis M. Ritchie, 2015
5. Oriented Object-Oriented Programming with C++ by Balaguruswamy, TMH

Suggested online courses (MOOCs)

1. NOC:An Introduction To Programming Through C++, IIT Bombay by Prof. Abhiram G Ranade https://nptel.ac.in/courses/106101208
2. Programming in Modern C++, IIT Kharagpur By Prof. Partha Pratim Das https://onlinecourses.nptel.ac.in/noc23_cs50/preview

This course can be opted as an elective by the students of following subjects: **BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **First**	Semester: **I**
Subject: Computer Science		
Course Code: **MCS-103N**	Course Title: **Data Structures**	
Course Objectives: The objective of the course is to familiarize students with basic data structures and their use in fundamental algorithms.		
Course Outcomes: **CO1:** Understand basic data structures such as arrays, strings, and linked lists. **CO2:** Study linear data structures such as stacks and queues and understand their difference. **CO3:** Describe the hash function and concepts of collision and its resolution methods. **CO4:** Study tree, heap and graphs along with their basic operations. **CO5:** Study different techniques for solving problems like sorting and searching		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**BLOCK - 1**
Unit 1	**Introduction to data structure:** Algorithm, Basic criteria for algorithms, Data type, Data structure, Data representation, linear and nonlinear data structure.
Unit 2	**Basics of algorithm:** Algorithm, Basics of complexity of algorithm
Unit 3	**Array:** Definition, Representation of array, Single and multi-dimensional array, address calculation (one dimensional, two dimensional, multidimensional), sparse matrices
Block 2	**BIOCK – 2**
Unit 4	**Stack:** Definition, Operations on stacks, Array representation and implementation of stack; infix, prefix and postfix representation of expression and evaluation multiple stacks, Application of stacks.
Unit 5	**Recursion:** Recursive definition and processes, some named problems of recursion, principle of recursion: designing recursive algorithm, how recursion works, tail recursion.
Unit 6	**Queue:** Definition, operation on queues, circular queue, dequeue, priority queue, Application of queue.
Block 3	**BLOCK 3**
Unit 7	**Linked List:** Representation and implementation of single linked list, Operations in the singly linked list, stack and queue as a linked list, circularly linked list, doubly linked list, circularly doubly linked list, Application of linked list: polynomial representation and addition, garbage collection
Unit 8	**Tree:** Basic terminology, binary tree, binary tree representation, complete binary tree, extended binary tree, array and linked list representations, traversing binary tree, threaded binary tree, binary search tree, Operations on BST, AVL tree, Operations on AVL tree, B-tree Insertion and deletion in B tree.
Unit 9	**Graph:** Basic terminology Graph representation Depth first search, breadth first search, topological sort, connected components, spanning tree, minimum cost spanning tree, Kruskal's and prim's algorithm, Shortest path algorithms: Bellman Ford Algorithm, Dijkstra's algorithm, Floyd-Warshall algorithm.
Block 4	**BLOCK- 4**
Unit 10	**Searching and sorting:** Sequential search, binary search, comparison and analysis, Selection sort, Bubble sort, Insertion sort, Heap sort, Quick Sort, Merge sort, Shell sort, radix sort.
Unit 11	**Hashing:** Hash table, hash function, collision resolution strategies, hash table implementation.
Unit 12	**File Structure:** Terminology, File organization, Sequential files, Direct File organization, Indexed Sequential file organization.

Suggested Readings:

1. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.
2. R.L. Kruse: Data Structures & Program Design in C, PHI.

Suggested online courses (MOOCs)

1. Programming and Data Structure, IIT Kharagpur by Dr. P.P.Chakraborty https://nptel.ac.in/courses/106105085
2. NOC:Programming and Data structures (PDS), IIT Madras by Dr. N S. Narayanaswamy https://nptel.ac.in/courses/106106130
3. NOC:Programming, Data Structures and Algorithms, IIT Madras by Prof. Hema A Murthy, Dr. N S. Narayanaswamy, Prof. Shankar Balachandran https://nptel.ac.in/courses/106106127
4. Data Structures And Algorithms, IIT Delhi by Prof. Naveen Garg https://nptel.ac.in/courses/106102064

This course can be opted as an elective by the students of following subjects: **B.Sc. in computer science, B.Sc. in Statistics, BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **First**	Semester: **I**
Subject: Computer Science		
Course Code: **MCS-104P**	Course Title: **Data Structures and C++ Lab**	

Course Objectives: The aim of this course is to enhance programming skills while improving their practical knowledge of data structures. It strengthens the practical ability to apply suitable data structures for real-time applications.

Course Outcomes:

CO1 Implement the abstract data type and reusability of a particular data structure.
CO2 Implement linear data structures such as stacks, queues using array and linked list.
CO3 Understand and implements non-linear data structures such as trees, graphs.
CO4 Implement various kinds of searching, sorting and traversal techniques and know when to choose which technique.

Credits: **04**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

List of Practical in Data Structures Lab with C++:

1. Implementation of Stacks, Queues (using both arrays and linked lists).
2. Implement a program to evaluate a given postfix expression using stacks.
3. Implement the following operations on singly and circular linked list: i) Creation ii) Insertion iii) Deletion iv) Traversal
4. Implementation of operations on binary tree (delete entire tree, copy entire tree, mirror image, level order, search for a node etc.)
5. Implementation of the following operations on binary search tree (BST): (a) Minimum key (b) Maximum key (c) Search for a given key (d) Delete a node with given key
6. Implementation of graph traversals by applying: (a) BFS (b) DFS
7. Implement the following algorithms to find out a minimum spanning tree of a simple connected undirected graph: (a) Prim‟s algorithm (b) Kruskal‟s algorithm
8. Implement Dijkstra‟s algorithm for solving single source shortest path problem.
9. Implementation of recursive and non-recursive functions to perform the following searching operations for a key value in a given list of integers: i) Linear search ii) Binary search
10. Implement the following sorting algorithms: a) Bubble sort b) Selection sort c) Insertion sort (d) Merge sort (e) Quick sort (f) Heap sort
11. Write a C++ program to illustrate the concept of class with method overloading.
12. Write a C++ Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util)
13. Write a C++ program to illustrate the concept of Single level and Multi level Inheritance.
14. Write a C++ program to demonstrate the Interfaces & Abstract Classes.
15. Write a C++ program to implement the concept of exception handling.

Suggested Readings:

1. Virtual Lab on Data Structure: https://ds1-iiith.vlabs.ac.in/

Programme: **Master of Science**	Year: **First**	Semester: **I**
Subject: **Computer Science**		
Course Code: **PGBR-01**	Course Title: **Basics in Research**	

Course Objectives:

- To discuss the ***Sources of information***
- To discuss about ***journal abbreviations***
- To discuss the ***monographs, dictionaries, text books etc.***

Course Outcomes:

CO1 Able to learn about how to get information of research.
CO2 Learn about journal and article and research manuals
CO3 Able to know the role of primary, secondary and tertiary sources of information.
CO4 Gain knowledge about abstract and citation index.
CO5 Also know about digital web resources

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

(Syllabi should be framed block wise/unit wise; No of blocks and units may change)

Unit I	**Literature Survey** Sources of information, need for reviewing literature, primary-secondary and tertiary sources, journals, journal abbreviations, abstracts, current titles, reviews, monographs, dictionaries, text books, current contents, patents. subject index, substance index, author index, formula index and other indices with examples. Digital: Web resources, E-journals, journal access, TOC alerts. Hot articles: Citation index, UGC infonet, E-books, Impact Factors, Search engines-Google scholar, Wiki-databases, Science Direct, SciFinder, Scopus.
Unit II	**Ethics and IPR** Regulatory bodies, practices and compliances, Good Laboratory Practices (GLP), Research Ethics and Misconduct, Patents, Copyrights, Trademarks, Product and process of patenting, Patent Treaties and Convention, process of filing patent, database of patent, search and retrieval.
	Suggested Text Book Readings: **1.** Use different searching engine to get relevant information (Google scholar, chemical industry, Wiki-databases, chem Spider, Science Direct, SciFinder, Scopus. **2.** Access to different online research library and research portal (Web resources, E-journals, journal access, TOC alerts)

Note:- In this paper, learner itself study the objectives and prepare a report. The report will be submitted along with assignment to respective study center for evaluation. The maximum marks for evaluation are 100.

Programme: **Master of Science**	Year: **First**	Semester: **II**
Subject: Computer Science		
Course Code: **MCS-106N**	Course Title: **Computer Organization**	
Course Objectives: The course aim to provide understanding the basic structure of a digital computer and to study the operations of internal components.		
Course Outcomes: **CO1** Assess basics components of computer hardware. **CO2** Understand how Boolean algebra is related to designing computer logic, through simple combinational and sequential logic circuits. **CO3** Realize a simple computer with hardware design including data format, instruction format, instruction set, addressing modes, bus structure, input/output, memory, Arithmetic/Logic unit, control unit, and data, instruction and address flow. **CO4** Design combinational and sequential logic circuits, flip-flops, counters, shift registers, adders, substractor, multiplexer, demultiplexer, Arithmetic/Logic unit. **CO5** Develop concept of memory unit and input/output architecture. **CO6** Build basics of Instruction Set Architecture (ISA).		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Introduction to Digital Electronics**
Unit 1	**Introduction to number system:** binary, octal, hexadecimal, Inter-conversion to different number system.
Unit 2	**Boolean algebra and Logic Gates:** De Morgan's theorem, Boolean Identity. OR, AND NOT NAND, NOR and Ex OR gates and their Truth Tables, Positive and Negative logic.
Unit 3	**Reduction Techniques:** Standard representation of Boolean expressions, SOP and POS forms, Combinational and sequential circuits, Minterm and Maxterm expressions, Map reduction techniques, K- tap. Code Conversions: Binary to Gray, BCD to decimal etc.
Unit 4	**Binary Arithmetic:** Half and Full Adder, Substractor, Multiplexer, Demultiplxer, Decoder, Encoders, Comparators.
Unit 5	**Sequential Circuit:** Flip Flops: S/R, J/K, D and T Latches, Digital Counters, Registers.
Block 2	**Basic building blocks**
Unit 6	**Building blocks:** I/O, Memory, ALU and its components, Control Unit and its functions
Unit 7	**Instruction** — word, Instruction and Execution cycle, branch, skip, jump and shift instruction, Operation of control. registers; Controlling of arithmetic operation.
Unit 8	**Addressing techniques —** Direct, Indirect, Immediate, Relative, Indexed addressing and paging. Registers —Indexed, General purpose, Special purpose, overflow, carry, shift, scratch, Memory Buffer register; accumulators; stack pointers; floating point; status information and buffer registers.
Block 3	**Memory & I/O**
Unit 9	**Memory**: Main memory, RAM, static and dynamic, ROM, EPROM, EEPROM, EAROM, Cache and Virtual memory.
Unit 10	**I/O System:** Buses, Interfacing buses, Bus formats- address, data and control, Interfacing keyboard, display, auxiliary storage devices and printers.
Unit 11	**Introduction to Microprocessors and microcontrollers**; Introduction to 8085 microprocessor, example of few instructions to understand addressing techniques,

	differences between microprocessors and microcontrollers. Interlocution to different processor families.

Suggested Readings:

1. William Stallings, "Computer Organization and Architecture", 9th Edition, PHI,2012
2. M. Morris Mano, Michael D. Ciletti, "Digital Design", 4th Edition, Pearson Education, 2011.
3. Hennessy J. and Patterson D., "Computer Architecture: A Quantitative Approach", 5th Edition, Morgan Kaufmann, 2011.

Suggested online courses (MOOCs)

1. Digital Computer Organization, IIT Kharagpur by Prof. P.K. Biswas
   https://nptel.ac.in/courses/117105078
2. NOC:Computer architecture and organization, IIT Kharagpur by Prof. Indranil Sengupta, Prof. Kamalika Datta
   https://nptel.ac.in/courses/106105163
3. NOC:Computer Organization and Architecture, IIT Madrasby Prof. V. Kamakoti
   https://nptel.ac.in/courses/106106166
4. Computer Organisation and Architecture, IIT Kanpurby Prof. Bhaskaran Raman
   https://nptel.ac.in/courses/106104073

This course can be opted as an elective by the students of following subjects: **B.Sc. in computer science, BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **First**	Semester: **II**
Subject: Computer Science		
Course Code: **MCS-108N**	Course Title: **Data Communication and Computer Networks**	
Course Objectives: This course offers students an understanding of how machines are connected in a network and how data communication takes place between machines at various locations. It provides basic concepts of data communication, layered model, protocols and interworking between computer networks and switching components in telecommunication systems.		
Course Outcomes: **CO1** Explain basics of OSI Reference Model and TCP/IP Model. **CO2** Understand basics of computer networks and various network topologies. **CO3** Understand various protocol of data link layer for flow and error control such as Stop and wait protocols, One bit sliding window protocol, Using Go-Back N. **CO4** Describe different types of network devices Hub, Bridges, Switch, Gateways, and Routers along with their working. **CO5** Realize how packet is being transferred from source to destination PC. **CO6** Understand the knowledge of network management and communication switching techniques.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Block 1	**Computer Networks Basics**	
	Introduction: Layered network architecture, Review of ISO-OSI Model. Data Communication techniques: Pulse code Modulation, (PCM), Data modems, Multiplexing techniques –Frequency-Division, Time-Division, Time-Division Transmission Media-Wires, Cables, Radio, Links, Fiber-Optic Links. Asynchronous Transfer Mode (ATM); Cell Format, Layovers in ATM, Class 1,2,3,4 Traffic Random Access Data Networks, Concept of Random Access, Pure ALOHA; Throughput Characteristics Slotted ALOHA, Throughputs for Finite and Infinite, Population S- ALOHAS. MARKOV Chain Model for S- ALOHAS. Throughputs for Finite and Infinite, Population S- ALOHAS. MARKOV Chain Model for S-ALOHA.	
Block 2	**Data Link layer**	
	Local Area Networks (LANs): IEEE 802.4 and 802.5 Protocols. Performance of Ethernet and Token ring protocols, FDDI Protocol, Distributed Queues Dual Bus (DQDB) Protocol. Data Link Protocols: Stop and Wait Protocols: Noise Free and Noisy Channels Performance and Efficiency, Verification of protocols using Finite State Marching. HDLC Data Link Protocol.	
Block 3	**Network & Transport Layer**	
	Network Layer Protocols: Design issue: Virtual circuits and Datagram. Integrated Services Digital Network: Interfaces, Devices, Channel Structure. Dead Locks and their avoidance Network Layer in ATM, Internetworking: Bridges, Routers and Gateways, Internet Architecture and Addressing. Transport Layer Protocols: Design issues: Quality of Services, Primitives Connection Management: Addressing, Connection Establishment and Releases, Use of Timers, Flow Control and Buffering, Multiplexing, Crash Recovery.	
Block 2	**Upper Layer Protocols**	

	Routing Algorithms: Optimality Principle, Shortest Path Routing- Dijkstra, Bellman – Ford and Floyd- War shall Algorithm. Elements of TCP/IP Protocol: User Datagram Protocol Connection Management, Finite State Machine. Session Layer Protocols: Dialog Management, Synchronization, OSI Session Primitives Connection Establishment, Presentation and Application Layer Protocols: Presentation Concepts NMP- Abstract Syntax Notation-1 (ASN-1), Structure of Management, Management Information Base.

Suggested Readings:

1. HBehrouz A. Forouzan, Data Communications and Networking, McGraw Hill , 2006
2. A.S. Tanenbaum, Computer Networks, PHI , 2002

Suggested online courses (MOOCs)

1. Data Communication, IIT Kharagpur by Prof. Ajit Pal
   https://nptel.ac.in/courses/106105082
2. NOC:Computer Networks and Internet Protocol, IIT Kharagpur by Prof. Soumya Kanti Ghosh, Prof. Sandip Chakraborty
   https://nptel.ac.in/courses/106105183
3. NOC:Advanced Computer Networks, IIT Indore, IIT Gandhi nagar by Prof. Neminath Hubballi, Prof. Sameer Kulkarni
   https://nptel.ac.in/courses/106106243

This course can be opted as an elective by the students of following subjects: **BCA, MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **First**	Semester: **II**
Subject: Computer Science		
Course Code: **MCS-109N**	Course Title: **Data Base Management System**	

Course Objectives: Today databases form the backbone of all major applications – internet, banking, product & sales etc. Relational Database Management Systems (DBMS) have long formed the basis for many leading databases such as Oracle, Microsoft SQL Server and MySQL. This course aim to provide a common set of models and design paradigms which includes:

- Data models, conceptualize and depict a database system using ER diagram.
- Internal storage structures in a physical DB design.
- Database normalization technique that organizes the data within a database in the most efficient manner possible.
- Fundamental concepts of transaction processing techniques**.**

Course Outcomes:

CO1 Students can explain the role of a database management system, basic database concepts, including the structure and operation of the relational data model.

CO2 Apply logical database design principles, including E-R/EE-R diagrams, conversion of ER diagrams to relations.

CO3 Describe the concepts of integrity constraints, relational algebra, relational domain & tuple calculus, data normalization.

CO4 Construct simple and moderately advanced database queries using Structured Query Language (SQL).

CO5 Understand and apply Database Normalization to remove the duplicate data and database anomalies from the relational table

CO6 Understand the concept of a database transaction including concurrency control, backup and recovery.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Basic concepts of DBMS**
Unit 1	**Introduction: Database** Management System, Examples, Characteristics of the Database Approach, Advantage of using a Database Approach. Database System concepts and Architecture, Data Models, Schemes and Instances, DBMS Architecture and Data independence, Database Languages, Procedural and Non-procedural languages and Interfaces. Database System Environment, Classification of Database Management Systems.
Unit 2	**ER Model:** Database Modeling using the ER Model., Using High-Level conceptual Data Models for Database design, an example Database Application, Entity types, Entity Sets, Attributes and keys, Relationships, Relationship types, roles and Structural Constraints., Week Entity types, Refining the ER Design for the Company Database, ER Diagrams, naming conventions and design Issues, Conversion of ER Diagram to tables.
Unit 3	**Relational Data Model:** Basic Relational data model Concepts, Relational Databases and Relational Database Schemas, Relational Model Constraints, update Operations and Dealing with Constraint Violations
Block 2	**Query Language and Database Design Concepts**
Unit 4	**Relational Algebra:** Relational Model Concepts, Relational concepts and Relational Database Schemes, Update Operation and Dealing with Constraints Violations, Relational Database Design, Using ER-to-Relational Mapping.
Unit 5	**Structured Query language:** Data definition, Constraints and Schema changes in SQL 2, Basic Quires in SQL, More Complex SQL Quires, Insert, Delete and Update Statements in SQL, views (Virtual Tables) in SQL, Specifying general constraints as Assertion features of SQL. Integrity constraints, Triggers, Functional dependencies.

Unit 6	**Functional Dependency Theory: Functional** Dependencies and Normalization for Relational Database, Informal Design Guidelines for Schemes, Functional Dependencies.
Unit 7	**Normalization:** Normal Forms based on Primary keys, General Definitions of Second and Third Normal forms, Boyce Codd Normal form, Relational Database Design Algorithms and Further Dependencies, Algorithms for Relational Database Schema Design, Multi-valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form, Inclusion Dependencies, Other Dependencies and Normal Forms.
Block 3	**Transaction Management & Emerging Databases**
Unit 8	**Transaction Processing Concepts**: Introduction to Transaction Processing, Transaction and System Concept, Desirable properties of Transactions, Scheduling and Recoverability, Serializability of Scheduling, Transaction Support in SQL, Concurrency control techniques, Concurrency techniques for concurrency control, concurrency control based on timestamp based protocol, validation based protocol, deadlock handling, Database Recovery Techniques based on Immediate Update, Failure classification, Shadow Paging, Log based recovery, failure with loss of Nonvolatile Storage.
Unit 9	Emerging Trends in DBMS: Emerging Trends in DBMS: Introduction to object-oriented Database Management System, Introduction to client/Server Database, Introduction to Distributed Database, Introduction to Knowledge Databases.

Suggested Readings:

1. R Elmasri, S Navathe, Fundamentals of Database Systems, 6th edition, Addison-Wesley, 2010.
2. R Ramakrishnan, J Gehrke, Database Management Systems, 3rd Ed., McGraw-Hill, 2002.
3. A Silberschatz, H Korth and S Sudarshan, Database System Concepts, 6th Ed., McGraw-Hill, 2010.

Suggested online courses (MOOCs)

1. NOC: Data Base Management System, IIT Kharagpur by Prof. Partha Pratim Das Prof. Samiran Chattopadhyay Prof. Kausik Datta
   https://nptel.ac.in/courses/106105175
2. NOC:Introduction to Database Systems, IIT Madras by Prof. P.Sreenivasa Kumar
   https://nptel.ac.in/courses/106106220
3. NOC:Fundamentals of Database Systems (Course sponsored by Aricent), IIT Kanpur By Dr. Arnab Bhattacharya
   https://nptel.ac.in/courses/106104135

This course can be opted as an elective by the students of following subjects: **B.Sc. in Computer Science, BCA, MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **First**	Semester: **II**
Subject: Computer Science		
Course Code: **MCS-110P**	Course Title: **Database Management Systems Lab**	

Course Objectives:

- Provide working on existing database systems, designing of database, creating relational database, analysis of table design.
- Practice various DDL commands in SQL
- Write simple and complex queries in SQL
- Familiarize PL/SQL

Course Outcomes:

CO1 Design and implement a database schema for a given problem

CO2 Populate and query a database using SQL and PL/SQL

Credits: **04**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

List of Practical in Database Management Systems Lab:

Creation of a database (exercising the commands for creation)

1. Simple to complex condition query creation using SQL Plus.
2. Implementation of DDL commands of SQL with suitable examples: Create table, Alter table and Drop Table
3. Implementation of DML commands of SQL with suitable examples: Insert, Update and Delete
4. Implementation of different types of function with suitable examples: Number function, Aggregate Function, Character Function, Conversion Function and Date Function
5. Implementation of different types of operators in SQL: Arithmetic Operators, Logical Operators, Comparison Operator, Special Operator and Set Operation.
6. Implementation of different types of Joins: Inner Join, Outer Join and Natural Join etc.
7. Study and Implementation of Group By, having clause, Order by clause and Indexing.
8. Implementation of Sub queries and Views.
9. Usage of triggers and stored procedures.
10. Writing PL/SQL procedures for data validation.

Suggested Readings:

1. https://www.cdlsiet.ac.in/wp-content/uploads/2022/03/DBMS-LAB-MANUAL.pdf
2. https://mrcet.com/pdf/Lab%20Manuals/CSE%20II-II%20SEM.pdf

Course prerequisites: To study this course, a student must have qualified graduation with Mathematics.		
Programme: M.Sc.	**Year:** I	**Semester:** II
Subject: Computer Science		
Course Code: *PGMP-02*	**Course Title:** *Mini Project*	
Course Objectives: In the second semester of Masters the main objectives of the exposure of students towards the project is to elevate their understanding into the applications areas of Mathematics. This course will develop their analytical ability, will provide them an apt exposure to work in any research group, and will motivate them to execute research in the area of their interest in Mathematical sciences.		
Course Outcomes: **CO1:** Students will be able to plan and strategize a scientific problem, and implement it within a reasonable time frame. **CO2:** It is expected that after completing this project dissertation, students will learn to work independently and how to keep accurate/readable record of assigned project. **CO3:** In addition, students will be able to know the library search and handle the data in a meaningful way. **CO4:** Subsequently, the students should be able to critically examine research articles, and improve their scientific writing/communication skills and power point presentation.		
Credits: 4	**Type of Course:** Core	
Max. Marks: 100	**Min. Passing Marks:** 36	
Topic	**Note:** Students shall make mini project on selected topic of their own choice studied so far (with or without any, Supervisor) and prepare the report. The report will be submitted along with assignment to respective study center for evaluation. The maximum marks for evaluation are 100.	
Suggested Text Book Readings: 1. Use different searching engine to get relevant information (*Google scholar, Wiki-databases, Science Direct, SciFinder, Scopus, and YouTube.* 2. Access to different online research library and research portal (*Web resources, E-journals, journal access, TOC alerts*)		

Programme: **Master of Science**	Year: **Second**	Semester: **III**
Subject: Computer Science		
Course Code: **MCS-111N**	Course Title: **Design And Analysis Of Algorithms**	

Course Objectives: This course provide the common paradigms to design efficient algorithms for real world problem solving. It gives an understanding of how to analyze the asymptotic performance of algorithm; write rigorous correctness proofs for algorithms; important algorithmic design paradigms and methods of analysis; efficient algorithms in common engineering design situations.

Course Outcomes:

CO1 Understand that various problem solving methods exist such as; iterative technique, divide and conquer, dynamic programming, greedy algorithms.

CO2 Analyze the strengths and weaknesses of an algorithm theoretically as well as practically.

CO3 Identify and apply an appropriate technique to design an efficient algorithm for simple problems.

CO4 Demonstrate correctness and efficiency of the algorithm.

CO5 Apply various searching and sorting algorithms.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Introduction and Design Strategies-I**
Unit 1	**Introduction:** Algorithm, Psuedo code for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Growth of functions: Asymptotic Notation, Recurrences: substitution method, master method.
Unit 2	**Divide and Conquer:** General method, applications-Binary search, Finding the maximum and minimum, Quick sort, Heapsort, Strassen's Matrix Multiplication.
Unit 3	**Sorting in Linear Time:** Lower bounds for sorting, Counting sort, Radix sort, Bucket sort, Medians and Order Statistics, Minimum and maximum.
Block 2	**Algorithm Design Strategies-II**
Unit 4	**Greedy method:** General method, applications- Knapsack problem, Job sequencing with deadlines, optimal two way merge patterns, Huffman codes, Minimum cost spanning trees: Prims and Kruskal's algorithm, Single source shortest paths: The Bellman-Ford algorithm, Dijkstra's algorithm.
Unit 5	**Dynamic Programming:** General method, applications, capital budgeting problem, Multistage graphs, Matrix chain multiplication, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem.
Block 3	**Algorithm design strategies & Completeness**
Unit 6	**Graph Algorithms:** Introduction, representation of graphs, Breadth first search, depth first search, topological sort, strongly connected component, flow networks, ford-fulkerson method.
Unit 7	**Backtracking:** General method, applications, 8-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.
Unit 8	**Branch-And-Bound:** The method, travelling salesperson problem, 15 puzzle problem.
Unit 9	**NP-Hard and NP-Complete problems:** Basic concepts, non-deterministic algorithms, NP - Hard and NP Complete classes, satisfiability problem, reducibility.

Suggested Readings:

1. Cormen, Leiserson, Rivest,and Stein, "Introduction to Algorithms", MIT Press ,Third Edition, 2009.
2. Dasgupta, Papadimitrou and Vazirani, "Algorithms", McGraw-Hill Education, 2006. Horowitz, Sahni, and Rajasekaran, "Computer Algorithms" Silicon Press, 2007

Suggested online courses (MOOCs)

1. NOC:Design and Analysis of Algorithms, Chennai Mathematical Institute By Prof. Madhavan Mukund

https://nptel.ac.in/courses/106106131 2. NOC:Introduction to algorithms and analysis, IIT Kharagpur by Prof. Sourav Mukhopadhyay https://nptel.ac.in/courses/106105164 3. Design and Analysis of Algorithms, IIT Bombay By Prof. Abhiram Ranade https://archive.nptel.ac.in/courses/106/101/106101060/#
This course can be opted as an elective by the students of following subjects: **MCA**
Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **Second**	Semester: **III**
Subject: Computer Science		
Course Code: **MCS-112N**	Course Title: **Java Programming**	
Course Objectives: This course aims to cover the essential topics of Java programming so that students can improve their skills to cope with the current demand of IT industries and solve many problems in their field of study.		
Course Outcomes: **CO1** Use the characteristics of an object-oriented programming language JAVA in a program. **CO2** Apply JAVA features to program design and implementation. **CO3** Design and implementation programs of Java Script, Applets, Event Handling, AWT Programming, and Interface. **CO4** Implementation of Packages, Swing, and Servlet. **CO5** Design and implementation programs of JSP.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Block 1	**Object Oriented Methodology and Java**	
Unit 1	**Object Oriented Programming:** Paradigms of Programming languages, Evolution of Object-Oriented Methodology, Basic Concepts of OO Approach, Comparison of object oriented and procedure - oriented Approaches, Benefits of OOPS, Applications of OOPS. Classes and objects, Abstraction and Encapsulation, Inheritance, Method overriding and Polymorphism.	
Unit 2	**Java Language Basics**: Introduction to Java, Primitive Data Type and Variables, Java Operators.	
Unit 3	**Expressions Statements and Arrays:** Expressions, Statements, Control Statements, Selection Statements, Iterative Statements, Jump statements, Arrays.	
Block 2	**Object oriented concepts and Exceptions Handling**	
Unit 4	**Class and objects:** Class Fundamentals, Introducing Methods, this Keyword, Using objects as Parameters, Method overloading, Garbage collection, the ffinalize () Method.	
Unit 5	**Inheritance and Polymorphism:** Inheritance Basics, Access, Multilevel, inheritance, Method overriding Abstract classes, Polymorphism, Final Keyword.	
Unit 6	**Packages and interfaces:** Package, Accessibility of Packages, using Package members, Interfaces, Implementing interfaces, interface and Abstract classes, Extends and Implements together.	
Unit 7	**Exceptions Handling:** Exception, Handling of Exception, Types of Exceptions, Throwing, Exceptions, writing Exception subclasses.	
Block 3	**Multithreading, I/O, and Strings Handling**	
Unit 8	**Multithreaded Programming:** Multithreading, The Main thread, JAVA Thread Model, Thread Priorities, Synchronization in JAVA, Inter thread Communication.	
Unit 9	**I/O In Java:** I/O Basics, Streams and stream, Classes, the predefined streams, Reading from and writing to console, reading and writing files, the transient and volatile Modifiers, using instance of Native Methods.	
Unit 10	**Strings and Characters:** Fundamental of Characters and Strings, the String class, String operations, Data Conversion using value of () Methods, Strings Buffer and Methods.	
Unit 11	**Exploring Java I/O:** Java I/O classes and interfaces, Stream classes, Text streams, Stream Tokenizer, Serialization, Buffered stream, print stream, Random Access file.	
Block 4	**Graphics and user interfaces**	
Unit 12	**Applets:** The applet class, Applet architecture, An applet Skeleton: Initialization and Termination, Handling events, HTML Applet TAG.	

Unit 13	**Graphics and user interfaces:** Graphics contests and Graphics objects, user interface components, Building user interface with AWT, Swing - Based GUI, Layouts and layouts and layout Manager, Container.
Unit 14	**Networking Features:** Socket overview, reserved parts and proxy servers, Internet Addressing: Domain Naming Services (DNS), Java and The Net: URL, TCP/IP Sockets, Datagrams.

Suggested Readings:

1. Java: The Complete Reference Hebert Schildt, Mc Graw Hill
2. Object-Oriented Programming with C++ and Java Debasis Samanta, Prentice Hall India.

Suggested online courses (MOOCs)

1. NOC:Programming in Java, IIT Kharagpur by Prof. Debasis Samanta: https://nptel.ac.in/courses/106105191

This course can be opted as an elective by the students of following subjects: **MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **Second**	Semester: **III**
Subject: Computer Science		
Course Code: **MCS-113N**	Course Title: **Operating System**	
Course Objectives: The course will introduce Operating Systems (OS), their design and implementation. We will discuss the goals of an OS and some successful and not-so-successful OS designs. We will also discuss the following OS services in detail: thread scheduling, security, process management, memory management, virtual memory, and disk scheduling.		
Course Outcomes: **CO1** Analyze & classify different types of operating system **CO2** Understand the working of Operating system **CO3** Interpret concepts of thread scheduling, process management, memory management, virtual memory, and disk scheduling.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**An Overview and Process Management**
Unit 1	**Introduction:** Basic definitions, Batch processing, Multi-programming. Time sharing, multiprocessing; Structure and Functions of Operating System
Unit 2	**Process and thread:** Process, Process states, State Transitions, Process Control Block, Context Switching, concept of thread, comparison between process and thread, Thread model, thread usage, implementing thread in kernel and user space.
Unit 3	**Process Scheduling:** Scheduler, Scheduling criteria, Preemptive and non-preemptive scheduling, Process Scheduling, Process scheduling algorithms.
Unit 4	**Concurrent Process:** Process Interaction, Shared Data and Critical Section, Mutual Exclusion, Synchronization, Classical Problems of Synchronization, Semaphores, Monitors.
Block 2	**Memory Management and Unix Case Study**
Unit 5	**UNIT 5: Deadlock:** Concept of deadlock, necessary condition for deadlock, resource allocation graph, deadlock prevention, deadlock avoidance, Banker's algorithm, Deadlock detection, deadlock recovery.
Unit 6	**UNIT 6: Memory management:** Address Binding, Dynamic Loading and Linking Concepts, Logical and Physical Addresses Contiguous and non-contiguous memory allocation, Paging, Segmentation, Virtual Memory, Demand Paging, Page fault,Page replacement algorithms, thrashing.
Unit 7	**UNIT 7: Secondary memory management:** Free Space management, Disk Structure, Disk Scheduling, Formatting, Swap space Management.
Unit 8	**UNIT 8: Case Study of UNIX**

Suggested Readings:

1. Silberschatz, Galvin, Gagne, Operating System Concepts, 8th Edition, Wiley,2008
2. Andrew S. Tanenbaum, Albert S. Woodhull, Operating Systems, Design and Implementation, 3rd Edition, Prentice Hall,2006.
3. William Stallings, Operating Systems: Internals and Design Principles, 6th Edition, Prentice Hall, 2013.
4. Charles Patrick Crowley, Operating Systems-A Design-oriented Approach. 1996

Suggested online courses (MOOCs)

1. NOC:Operating System Fundamentals, IIT Kharagpur by Prof. Santanu Chattopadhyay https://nptel.ac.in/courses/106105214
2. NOC:Introduction to Operating Systems, IIT Madras by Prof. Chester Rebeiro https://nptel.ac.in/courses/106106144
3. Operating Systems, IIT Delhi by Prof. Sorav Bansal

https://nptel.ac.in/courses/106102132
This course can be opted as an elective by the students of following subjects: **B.Sc. (Computer Science), BCA and MCA**
Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **Master of Science**	Year: **Second**	Semester: **III**
Subject: Computer Science		
Course Code: **MCS-115P**	Course Title: **Java Programming and Algorithm Lab**	

Course Objectives:

- Provide the concept of classes, inheritance and abstract classes.
- Prepare students to excel in object oriented programming and to succeed as a Java developer.
- Provide students with a solid foundation in OOP fundamentals required to solve programming problems.
- Inculcate multidisciplinary approach and an ability to relate java programming issues to broader application context.

Course Outcomes:

CO1 Understand the necessity for Object Oriented Programming paradigm over structured programming.
CO2 Develop java programs, analyze, and interpret object-oriented data and report results.
CO3 Demonstrate an ability to design an object-oriented system, AWT components.

Credits: **04**	Type of Course: **Practical Lab**
Max. Marks: **100**	Min. Passing Marks: 36

List of Practical in Java Programming and Algorithm Lab:

1. Write a java program for Method overloading and Constructor overloading.
2. Write a java program to display the employee details using Scanner class.
3. a) Write a java program to represent Abstract class with example.
   b) Write a java program to implement Interface using extends keyword.
4. Write a java program to implement method overloading, method overriding, dynamic method dispatch.
5. Write a java program to implement single, multilevel, hierarchal, multiple, hybrid inheritances.
6. Write java programs that demonstrate the use of abstract, this, super, static, final keywords.
7. a) Write a java program for creating a package and using a package.
   b) Write a java program to demonstrate the use of wrapper classes.
8. a) Write a java program using all five keywords of exception handling mechanism.
   b) Write a java program for creating customized (user) exception
9. a) Write a java program to create the following AWT components: Button, Checkbox, Choice, and List.
   b) Write java programs to create AWT application using containers and layouts.
10. a) Write a java program to create a file, write the data and display the data.
    b) Write a java program that reads a file name from user and displays its information.

Suggested Readings:

https://mrcet.com/pdf/Lab%20Manuals/Lab%20Manual%20Object%20Oriented%20Programming%20through%20JAVA.pdf

<table>
<tr><td colspan="2">Programme: Master of Science</td><td>Year: II</td><td>Semester: III</td></tr>
<tr><td colspan="4">Subject: Computer Science</td></tr>
<tr><td colspan="2">Course Code: PGRT-03</td><td colspan="2">Course Title: Basic Research Tools</td></tr>
<tr><td colspan="4">Course Objectives:
➢ To discuss the application of MS office
➢ To discuss different research tools for research work.
➢ To discuss application of softwares.
➢ To discuss about reference management tools</td></tr>
<tr><td colspan="4">Course Outcomes:
CO1 Able to learn about basic computer application of research work.
CO2 Learn about Latex tools with MS-XL
CO3 Able to know the role of Chem-Draw, Origin, SPSS, R-software, Octave, Matlab
CO4 Gain knowledge about application of Mendeley-software.
CO5 Also know about RefWorks and Zotero, etc.</td></tr>
<tr><td colspan="2">Credits: 04</td><td colspan="2">Type of Course: Core</td></tr>
<tr><td colspan="2">Max. Marks: 100</td><td colspan="2">Min. Passing Marks: 36</td></tr>
<tr><td>Unit I</td><td colspan="3">Application of MS Office/ Latex in research
Uses and application of MS Office/ Latex Tools with MS-XL, Power point Presentation.</td></tr>
<tr><td>Unit II</td><td colspan="3">Application of Softwares
Uses and application of Softwares such as plagiarism software, Statistical softwares, R-software, Matlab.</td></tr>
<tr><td>Unit III</td><td colspan="3">Reference management tools
Uses and application of Mendeley-software, EndNote, RefWorks and Zotero.</td></tr>
<tr><td></td><td colspan="3">Suggested Text Book Readings:
1. Microsoft office: Microsoft Office Essentials - IT Essentials: a Practical Guide - Subject Guides at University of York
2. How to Convert an Excel Table to a Latex table: How to Convert an Excel Table to a Latex table - YouTube
3. SPSS – What Is It: SPSS - Quick Overview & Beginners Introduction (spss-tutorials.com)
4. Video Processing in MATLAB: Video Processing in MATLAB - Video - MATLAB & Simulink (mathworks.com)</td></tr>
<tr><td colspan="4">Note:- In this paper, learner itself study the objectives and prepare a report. The report will be submitted along with assignment to respective study center for evaluation. The maximum marks for evaluation are 100.</td></tr>
</table>

Programme: **Master of Science**	Year: **Second**	Semester: **IV**
Subject: Computer Science		
Course Code: **MSCDS -117N**	Course Title: **Soft Computing**	

Course Objectives: Expose students to Neural Network, Fuzzy Logic and Genetic Algorithms, which are the major building blocks of Intelligent Systems.

Course Outcomes:

CO1–Discuss the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human experience.

CO2 –Understand how neural networks learn from available examples and generalize to form appropriate rules for inference systems.

CO3 –Provide the mathematical background for carrying out the optimization associated with neural network learning.

CO4 –Apply genetic algorithms and other random search procedures for finding global optimum of optimization problems.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Artificial Intelligence & Soft Computing:** Introduction of Artificial Intelligence, Problem domain of AI, AI techniques, Rule based system, monotonic reasoning, non-monotonic reasoning, Uncertainty reasoning & Inference, Bayesian theory and dependency network, Limitation of AI, Soft computing paradigms, pattern classification, association and mapping, Pattern recognition techniques.
Block 2	**Fuzzy Set Theory**: Introduction to Neuro – Fuzzy and Soft Computing – Fuzzy Sets – Basic Definition and Terminology – Set-theoretic Operations – Member Function Formulation and Parameterization – Fuzzy Rules and Fuzzy Reasoning – Extension Principle and Fuzzy Relations – Fuzzy If-Then Rules – Fuzzy Reasoning – Fuzzy Inference Systems.
Block 3	**Neural Network:** Structure and Function of a single neuron: Biological neuron, artificial neuron, definition of ANN, Taxonomy of neural net, Difference between ANN and human brain, characteristics and applications of ANN, single layer network, Perceptron training algorithm, Linear separability, Widrow & Hebb's learning rule/Delta rule, ADALINE, MADALINE, AI v/s ANN. Introduction of MLP, different activation functions, Error back propagation algorithm, derivation of BBPA, momentum, limitation, characteristics and application of EBPA, Deep Learning: Convolution Neural Network, Recurrent Neural Network.
Block 4	**Genetic Algorithm:** Fundamentals, basic concepts, working principle, encoding, fitness function, reproduction, Genetic modeling: Inheritance operator, cross over, inversion & deletion, mutation operator, Bitwise operator, Generational Cycle, Convergence of GA, Applications &advances in GA, Differences & similarities between GA & other traditional method.

Suggested Readings:

1. J.S.R.Jang, C.T.Sun and E.Mizutani, "Neuro-Fuzzy and Soft Computing", PHI, 2004, Pearson Education 2004.
2. S. Rajasekaran and G.A.VijaylakshmiPai.. Neural Networks Fuzzy Logic, and Genetic Algorithms, Prentice Hall of India.
3. Timothy J.Ross, "Fuzzy Logic with Engineering Applications", McGraw-Hill, 1997.
4. Davis E.Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley,N.Y.,1989.
5. S. Rajasekaran and G.A.V.Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI, 2003.
6. R.Eberhart, P.Simpson and R.Dobbins, "Computational Intelligence - PC Tools", AP Professional, Boston, 1996.

Suggested online courses (MOOCs)

1. NOC:Introduction to Soft Computing, IIT Kharagpur by Prof. Debasis Samanta https://nptel.ac.in/courses/106105173

This course can be opted as an elective by the students of following subjects: **M.Sc. (Statistics) and M.Sc. (Mathematics)**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **Master of Science**	Year: **Second**	Semester: **IV**
Subject: Computer Science		
Course Code: **MCS-121D**	Course Title: **Dissertation with viva voce**	
Course Objectives: ➢ To facilitate the learner to independently formulate and solve a social, philosophical, commercial, or technological problem and present the results in written and oral form. ➢ To render learners to real-life problems. ➢ To provide opportunities for learners to interact with people and present them confidently.		
Course Outcomes: **CO1** Investigate and evaluate a research topic relevant to environment and society. **CO2** Learn systematic discovery and critical review of appropriate and relevant information sources. **CO3** Apply qualitative and/or quantitative evaluation processes to original data. **CO4** Communicate research concepts and contexts clearly and effectively both in writing and orally		
Credits: **04**	Type of Course: **Research**	
Max. Marks: **100**	Min. Passing Marks:	

Programme: **Master of Science**	Year: **Second**	Semester: **IV**
Subject: Computer Science		
Course Code: **MCS-116N**	Course Title: **Computer Graphics**	
Course Objectives: The primary role of computer graphics is to render the digital content (0's and 1's) in a human-comprehensible form on the computer screen. This course introduces various object representation techniques along with 2D and 3D transformation, clipping, splines, objects modeling, colour modeling, lighting, textures and visible surface detection.		
Course Outcomes: **CO1** Demonstrate an understanding of contemporary graphics hardware. **CO2** Draw graphics using line & polygon and ability to perform operations on computer graphics. **CO3** Understand and demonstrate geometrical transformations, Segment, Windowing and Clipping, Interaction. **CO4** Demonstrate Hidden Surfaces & Lines; Light, Colour & Shading; Curves and Fractals		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Block 1	**Raster Graphics and Clipping**	
Unit 1	Introduction to Computer Graphics: What is Computer Graphics?, Application of Computer Graphics, Presentation Graphics, Painting and Drawing, Photo Editing, Scientific Visualization, Image Processing, Digital Art, Education, training, Entertainment and CAD Simulation, Animation and Games, Graphics Hardware, Input and Output Devices, Touch Panel, Light Pens, Graphic Tablets, Plotters, Film Recorders, Display Devices, Refreshing Display Devices: Raster-Scan, Random-Scan, Plasma Panel and LCD panels	
Unit 2	Graphics Primitives: Points and Lines, Line-drawing Algorithms: DDA Algorithm, Bresenham's line Algorithm, Circle-generating Algorithm: Properties of Circles, Midpoint Circle of Algorithm, Polygon Filling Algorithm: Scan-Line	
Unit 3	2-D Viewing and Clipping: Point Clipping, Line Clipping: Cohen-Sutherland Line Clippings, Cyrus-Beck Line Clipping Algorithm, Polygon Clipping: Sutherland Hodgman Algorithm, Windowing Transformation	
Block 2	**Transformations**	
Unit 4	2-D and 3-D Transformations: Basic Transformations: Translation, Rotation, Scaling, Shear, Composite Transformations: Rotations about a point, Reflection about a line, Homogeneous Coordinate Systems, 3-D Transformations	
Unit 5	Viewing Transformation: Projections: Parallel Projection, Orthographic & Oblique Projections, Isometric Projections, Perspective Projections	
Block 3	**Modeling & Rendering**	
Unit 6	Curves and Surfaces: Polygon Representation Methods: Polygon Surfaces, Polygon Tables, Plane Equations, Polygon Meshes, Bezier Curves and Surfaces: Bezier Curves, Properties of Bezier Curves, Bezier Surfaces, Surface of Revolution	
Unit 7	Visible – Surface Detection: Depth Buffer Method, Scan-Line Method, Area-Subdivision Method	
Unit 8	Polygon Rendering and Ray Tracing Methods: Illumination Model: Ambient Reflection, Diffuse Reflection, Specular Reflection, Shading: Gouraud Shading, Phong Shading, Ray Tracing: Basic Ray-Tracing Algorithm	
Suggested Readings: 2. J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Computer Graphics - Principles		

and Practice, Second Edition in C, Pearson Education, 2003.
3. D. Hearn and M. Pauline Baker, Computer Graphics (C Version), Pearson Education, 2nd Edition, 2004.
4. Edward Angel, Interactive Computer Graphics A Top-Down Approach with OpenGL 5th Edition, Addison-Wesley, 2008.
5. Prabat K Andleigh and KiranThakrar, "Multimedia Systems and Design", PHI, 2003.

Suggested online courses (MOOCs)

1. Computer Graphics, IIT Madras by Prof. Sukhendu Das https://nptel.ac.in/courses/106106090
2. Introduction to Computer Graphics, IIT Delhi by Prof. Prem K Kalra https://nptel.ac.in/courses/106102065
3. NOC:Computer Graphics, IIT Guwahati by Prof. Samit Bhattacharya https://nptel.ac.in/courses/106103224

This course can be opted as an elective by the students of following subjects: **B.Sc. (Computer Science) and BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **Master of Science**	Year: **Second**	Semester: **IV**
Subject: Computer Science		
Course Code: **MCS-114N**	Course Title: **Multimedia Technology**	

Course Objectives: Today, Multimedia and web design technology play an essential role in education, agriculture, product launch, science and technology, corporate development and enhanced business opportunities. The increasing variety of hardware and software components in multimedia and website design has escalated the demand for human resources in these fields. This course is designed to inculcate required skills for these activities.

Course Outcomes:

CO1 Visualize scopes of multimedia and understand steps in creation of multimedia applications.

CO2 Understand digital audio, Prepare audio required for a multimedia system and Speech synthesis and recognition concept.

CO3 Analyze representation of video, how video work and different video formats.

CO4 Describe different animation techniques and software used for animation.

CO5 Understand various multimedia development and authoring tools.

CO6 Know the different layers of network along with video conferencing technique.

Credits: **04**	Type of Course: **Core**
Max. Marks: **100**	Min. Passing Marks: **36**

Block 1	**Introduction to Multimedia and Its Components**
Unit 1	**Multimedia Technology:** Meaning & scope of Multimedia; Elements of Multimedia; Creating multimedia applications; Multimedia file & I/O functions; Multimedia data structures; Multimedia file formats; Multimedia Protocols
Unit 2	**Multimedia Audio:** Digital sound; Audio compression & decompression; Companding: ADPCM compression; MPEG audio compression; True Speech; Special effects and Digital Signal Processing: Audio synthesis; FM synthesis: Sound blaster card; Special effect processors on sound cards; Wave table synthesis; MIDI functions; Speech synthesis & Recognition
Unit 3	**Multimedia Video:** Representation of Digital video; Video capture: Frame grabbing; Full motion video; Live video in a window; Video processor; Video compression & decompression; Standards for video compression & decompression; Playback acceleration methods
BLOCK-2	**Multimedia Animation, Authoring Tools and Internet**
Unit 4	**Creating Multimedia Animation:** Icon animation; Bit-map animation; Real-time vs Frame by Frame animation; Object modeling in 3D animation; Motion control in 3D animation; Transparency; Texture. Shadows, Anti-aliasing; Human modeling & Animation; Automatic motion control
Unit 5	**Multimedia Authoring Tools:** Project editor; Topic editor; Hot-spot editor; Developing a multimedia title; Multimedia text authoring systems; Usage of authoring tools
Unit 6	**Multimedia on LANs & Internet:** Multimedia on LAN; Fast modems & Digital networks for multimedia; High speed digital networks; Video conferencing techniques; Multimedia interactive applications on Internet: Future Directions.

Suggested Readings:

1. "Li, Ze-Nian, Mark S. Drew, and Jiangchuan Liu. Fundamentals of multimedia. Upper Saddle River (NJ) Pearson Prentice Hall, 2004.
2. Jeffcoate, Judith. Multimedia in practice: technology and applications. Prentice-Hall, Inc., 1995.
3. Vaughan, Tay. Multimedia: Making it work. Tata McGraw-Hill Education, 2006.

4. Melliar-Smith, Peter Michael, and Louise E. Moser. "Multimedia Networking: Technology, Management and Applications. Hershey, PA Idea Group, 2002.

Suggested online courses (MOOCs)

1. Multimedia processing, IIT Kharagpur by Prof. Somnath Sengupta
   https://nptel.ac.in/courses/117105083
2. CIT-003: Web Based Technologies and Multimedia Applications
   By Prof. P. V. Suresh | Indira Gandhi National Open University
   https://onlinecourses.swayam2.ac.in/nou20_cs05/preview

This course can be opted as an elective by the students of following subjects: **B.Sc. (Computer Science) and BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **Master of Science**	Year: **Second**	Semester: **IV**
Subject: Computer Science		
Course Code: **MCS-119N**	Course Title: **Information and Network Security**	
Course Objectives: This course aims to provide a basic understanding of the existing algorithms used to protect users online and understand some of the design choices behind these algorithms. The course offers a workable knowledge of the mathematics used in cryptology. The course emphasizes giving a basic understanding of previous attacks on cryptosystems to prevent future attacks.		
Course Outcomes: **CO1** Identify information security goals, classical encryption techniques and acquire fundamental knowledge on the concepts of finite fields and number theory. **CO2** Understand, compare and apply different encryption and decryption techniques to solve problems related to confidentiality and authentication **CO3** Apply the knowledge of cryptographic checksums and evaluate the performance of different message digest algorithms for verifying the integrity of varying message sizes **CO4** Apply different digital signature algorithms to achieve authentication and create secure applications **CO5** Apply network security basics, analyze different attacks on networks and evaluate the performance of firewalls and security protocols like SSL, IPSec, and PGP. **CO6** Apply the knowledge of cryptographic utilities and authentication mechanisms to design secure applications.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Information security and Symmetric Ciphers**
Unit 1	**Introduction:** History, what is Information Security; Characteristics of Information; Information Security Model; Components of an Information Security; Aspects of Information security: Security attacks, Security Mechanism, and Security Services (X.800), Model for Network Security.
Unit 2	**Classical Encryption Techniques: Historical** background, symmetric cipher model, Substitution techniques, Transposition techniques, steganography.
Unit 3	**Block ciphers and DES:** Block cipher principles, Data encryption standard, strength of DES, differential and cryptanalysis, block cipher design principles, block cipher mode of operation.
Unit 4	**Confidentiality Using Symmetric Ciphers:** Placement of encryption function, traffic confidentiality, key distribution, random number generation.
Block 2	**Public key Encryption and Hash Functions**
Unit 5	**Introduction to Number Theory:** Prime numbers, Fermat's and Euler's theorem, discrete logarithm
Unit 6	**Public Key Cryptography:** Public-Key Cryptography Principles, RSA, Key Management: Diffi-Hellman key exchange.
Unit 7	**Message Authentication and Hash Functions:** Authentication requirements, Authentication Functions, Message Authentication codes, Hash Functions, SHA-1, MD5.
Unit 8	**Digital Signatures:** Digital signatures, Authentication protocols, Digital Signature standard
Block 3	**Network Security Applications**
Unit 9	**Authentication Applications:** Kerberos Motivation, X.509 authentication service
Unit 10	**Electronic Mail Security:** PGP: PGP Notation, PGP Operational Description, S/MIME
Unit 11	**IP Security: IP** Security Overview, IP Security Architecture, Authentication Header
Unit 12	**Web Security:** Web Security Threats, Web Traffic Security Approaches, Overview of Secure Socket Layer and Transport Layer Security, Overview of Secure Electronic Transaction
Block 4	**Intruders and Viruses**
Unit 13	**Intruders:** Intruders, Intrusion Techniques, Password Protection, Password Selection Strategies, Intrusion Detection,

Unit 14	**Malicious Programs:** Malicious Programs, Nature of Viruses, Types of Viruses, Macro Viruses, Antivirus Approaches
Unit 15	**Firewall**: Firewall Characteristics, Types of Firewalls, Firewall Configuration

Suggested Readings:

1. Douglas Stinson, "Cryptography Theory and Practice", 2nd Edition, Chapman & Hall/CRC.
2. B. A. Forouzan, "Cryptography & Network Security", Tata Mc Graw Hill.
3. W. Stallings, "Cryptography and Network Security", Pearson Education.

Suggested online courses (MOOCs)

1. NOC:Cryptography And Network Security, IIT Kharagpur by Prof. Sourav Mukhopadhyay https://nptel.ac.in/courses/106105162
2. Cryptography and Network Security, IIT Kharagpur by Dr. Debdeep Mukhopadhyay https://nptel.ac.in/courses/106105031

This course can be opted as an elective by the students of following subjects: **MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

Programme: **Master of Science**	Year: **Second**	Semester: **VI**
Subject: Computer Science		
Course Code: **MCS-104N**	Course Title: **Software Engineering**	
Course Objectives: Provide the current software engineering techniques and examine the software life-cycle, including software specification, design implementation, testing and maintenance. It presents software engineering methodologies for the development of Quality, cost-effective, schedule meeting software.		
Course Outcomes: **CO1** Describe software engineering layered technology and process framework. **CO2** Introduces theories, models, and techniques that provide a basis for the software development life cycle. **CO3** Introduces software testing approaches including verification and validation, static analysis, reviews, inspections, and audits. **CO4** Understanding of the role of project management including planning, scheduling, risk management, etc. **CO5** Work as an individual and/or in team to develop and deliver quality software.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Unit 1	**Software Engineering Fundamentals:** Definition of Software, Software characteristics, Software Applications. Software Process: Software Process Models - Waterfall model, prototyping model, spiral model, incremental model, concurrent development model. Project management Concepts: The Management Spectrum - The People, The Product, The Process, The Project.	
Unit 2	**Software Process and Project Metrics :** Measures , Metrics and Indicators , Software measurement Size -Oriented Metrics , Function - Oriented Metrics , Extended Function point metrics Software Project Planning : Project Planning Objectives , Software Project Estimation , Decomposition Techniques - Problem Based Estimation Process Based Estimation ,Empirical Estimation Models- The COCOMO Model Risk Analysis and Management: Software risks, Risk identification, Risk Projection, Risk Refinement, Risk Mitigation , Monitoring and Management.	
Unit 3	**Software Quality Assurance:** Basic concepts- Quality, Quality Control, Quality Assurance, Cost of Quality, Software Quality Assurance (SQA), Formal Technical Review Software Configuration Management: Baselines, Software Configuration Items, The SCM Process, Version Control, Change Control, Configuration Audit, Status Reporting. Analysis Concepts and Principles: Requirements Elicitation for Software, Analysis Principles. The Information Domain, Modeling, Partitioning, Essential and Implementation Views, Specification: Specification Principles, Representation, The Software Requirement Specification (SRS)	
Unit 4	**Design Concepts and Principles:** Design Principles, Design Concepts — Abstraction, Refinement, Modularity, Software Architecture, Control Hierarchy, Structural Partitioning, Data Structure. Software Procedure, Structure, Information Hiding, Effective Modular Design- Cohesion, Coupling Software Testing: Testing Objectives & principles, Unit Testing, Integration Testing (Top-Down Integration, Bottom. Up Integration, Regression Testing, Smoke Testing), Validation Testing (Alpha and Beta Testing), System Testing (Recovery Testing, Security Testing, Stress Testing, Performance Testing).	

Unit 5	**Reengineering:** Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering CASE Tools: What is CASE, Building Blocks of CASE, A Taxonomy of CASE Tools, Integrated CASE Environments, The integration Architecture, The CASE Repository.

Suggested Readings:

1. Mall, Rajib. Fundamentals of software engineering. PHI Learning Pvt. Ltd., 2018.
2. R.S. Pressman, Software Engineering – A Practitioner's Approach, 6th Edition, TMH, 2013.
3. Ian Sommerville, Software Engineering, 8th Edition, Addison Wesley, 2009.
4. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing, 2010.

Suggested online courses (MOOCs)

1. NOC:Software Engineering, IIT Kharagpur by Prof. Rajib Mall https://nptel.ac.in/courses/106105182
2. Software Engineering, IIT Bombay by Prof. Rushikesh K Joshi, Prof. Umesh Bellur, Prof. N.L. Sarda https://nptel.ac.in/courses/106101061

This course can be opted as an elective by the students of following subjects: **BCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **Second**	Semester: **IV**
Subject: Computer Science		
Course Code: **MCS-107N**	Course Title: **Theory of Computation**	
Course Objectives: The aim of this course is to introduce students with the mathematical model of machines. The course familiarize students with the concept of formal language, their relationships and corresponding automaton. It builds core concepts to design grammars and recognizers for different formal languages; identify ambiguity in grammar.		
Course Outcomes: **CO1** Understand what automata is and what its use are. **CO2** Analyze regular grammar and design finite automata for various regular languages. **CO3** Analyze context free grammar and design pushdown automata for different types of context free languages. **CO4** Compare and analyze different languages, grammars and machines. **CO5** Design Turing machine for unrestricted grammar (type 0). **CO6** Understand undecidable problems that cannot be solved using computers.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	

Block 1	**Regular Expression and Finite Automata**
Unit 1	**Alphabet, Strings and Languages:** Set, Relations, Alphabet, Strings, Languages, Finite Representation of Languages, Chomasky Hierarchy
Unit 2	**Finite Automata**: Finite State Systems, Basic Definitions Non-Deterministic finite automata (NDFA), Deterministic finite automata (DFA), Equivalence of DFA and NDFA, Finite automata with epsilon transitions, Removal of epsilon transitions.
Unit 3	**Regular Expressions**: Regular Expressions-Definition, Algebraic Laws of RE, Finite Automata and Regular expressions, Conversion from RE to FA, Conversion from FA to RE, Arden's Theorem.
Unit 4	**Introduction to Machines**: Concept of basic Machine, Properties and limitations of FSM, Moore and mealy Machines, Equivalence of Moore and Mealy machines. Minimization of DFA.
Unit 5	**Block 2 Context Free Grammar**
Block 2	**Properties of Regular Language:** The Pumping Lemma for Regular Sets, Applications of the pumping lemma, Closure properties of regular sets.
Unit 6	**Context Free Grammar:** Context Free Grammar (CFG)-Formal definition, sentential forms, leftmost and rightmost derivations, the language of CFG.
Unit 7	**Normal Forms:** Simplifications of CFG's- Removal of Useless Symbols, Removal of epsilon and Unit Production, Normal Forms-CNF and GNF.
Unit 8	**Context Free Languages (CFL):** Closure Properties of CFL, Decision Properties of CFL, Application of CFG, Pumping Lemma for CFL.
Block 3	**Block 3 Pushdown Automata and Turing Machine**
Unit 9	**Push Down Automata:** Formal Definition of Pushdown Automata, Pushdown Automata accepted by final state and empty state, Equivalence between CFG and PDA.

Unit 10	**Turing Machine:** Turing Machine (TM) –Formal Definition and behavior, Transition diagram, Instantaneous Description, Language of a TM, Variants of TM, Universal Turing Machine, Halting Problem, Church Thesis.
Unit 11	**Undecidability:** Recursive enumerable, Undecidable Problem About Turing Machines, Unsolvable Problems.

Suggested Readings:

1. Hopcroft and Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education, 3rd edition, 2006
2. Linz, Peter, and Susan H. Rodger. An introduction to formal languages and automata. Jones & Bartlett Learning, 2022.

Suggested online courses (MOOCs)

1. NOC:Introduction to Automata, Languages and Computation, IIT Kharagpur by Prof. Sourav Mukhopadhyay
   https://nptel.ac.in/courses/106105196
2. Formal Languages and Automata Theory, IIT Guwahati by Dr. Diganta Goswami, Dr. K.V. Krishna
   https://nptel.ac.in/courses/111103016
3. Theory of Automata, Formal Languages and Computation, IIT Madras by Prof. Kamala Krithivasan
   https://nptel.ac.in/courses/106106049
4. NOC:Theory of Computation, IIT Kanpur by Prof. Raghunath Tewari
   https://nptel.ac.in/courses/106104148

This course can be opted as an elective by the students of following subjects: **BCA, MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A

Programme: **Master of Science**	Year: **Second**	Semester: **IV**
Subject: Computer Science		
Course Code: **MCS-120N**	Course Title: **System Software**	
Course Objectives: This course aims to illustrate the working of the various phases of a general-purpose compiler. It explains the principles involved in compiler design. It will cover all the basic components of a compiler, along with machine code generation and optimizations.		
Course Outcomes: **CO1:** Understand design issues of a lexical analyzer and use of Lex tool **CO2:** Explain code generation and code optimization schemes **CO3:** Understand the working of linkers and loaders and other development utilities. **CO4:** Design structure of Assembler and macro processor for a hypothetical simulated computer.		
Credits: **04**	Type of Course: **Core**	
Max. Marks: **100**	Min. Passing Marks: **36**	
Block 1	**Introduction to System Software and software tools**	
Unit 1	**Language Processors:** Introduction, Language Processing Activities, Fundamentals of Language Processing & Language Specification, Language Processor Development Tools.	
Unit 2	**Data Structures for Language Processing:** Search Data structures, Allocation Data Structures.	
Unit 3	**Software Tools:** Software Tools for Program Development, Editors, Debug Monitors, Programming Environments, and User Interfaces.	
Unit 4	**Assemblers:** Elements of Assembly Language Programming, A Simple Assembly Scheme, Pass Structure of Assemblers, Design of a Two Pass Assembler, A single pass Assembler for IBM PC.	
Unit 5	**Macro Processors:** Macros and Macro Processors: Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design of a Macro Preprocessor.	
Block 2	**Compilers and Interpreters**	
Unit 6	**Lexical Analysis:** Introduction to NFA and DFA, Lexical Analysis: Role of a Lexical analyzer, input buffering, specification and recognition of tokens, Finite Automata, Designing a lexical analyzer generator, Pattern matching based on NFA's.	
Unit 7	**Compiler- Syntax Analysis:** Syntax Analysis: Role of Parser, Top-down parsing, recursive descent and predictive parsers (LL), Bottom-Up parsing, Operator precedence parsing, LR, SLR and LALR parsers. (First and follow technique for generating a parse table is to be taught), Phases of the Compiler, Aspects of compilation, Memory allocation. Compilation of expressions and control structures.	
Unit 8	**Compiler- Code Generation:** Intermediate languages: graphical representations, DAGs, three address code, types of three address statements, syntax directed translation into three address code, implementation of three address statements.	
Unit 9	**Compiler- Optimization** Code Optimization: Machine dependent and machine independent code generation: Sources of optimization-Code Generation-Semantic stacks, evaluation of expressions, control structures, and procedure calls.	
Unit 10	**Interpreters:** Use and overview of interpreters, pure and impure interpreters	
Block 3	**Linker, Loaders and device Drivers**	

Unit 11	**Loaders and Linkers**: Basic loader functions: Design of an Absolute Loader – A Simple Bootstrap Loader, Machine dependent loader features Relocation – Program Linking – Algorithm and Data Structures for Linking Loader. Machine-independent loader features – Automatic Library Search – Loader Options Loader design options – Linkage Editors – Dynamic Linking – Bootstrap Loaders. Implementation examples: MSDOS linker.
Unit 12	**Device drivers:** Design and anatomy of UNIX device driver, Types of device driver, General design of UNIX character device driver, General design of UNIX block device driver, UNIX device driver installation.

Suggested Readings:

1. Alfred V. Aho, Jeffrey D Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education Asia, 2008
2. K.D. Cooper, and L. Torczon,Engineering a Compiler,Elsevier, 2004.

Suggested online courses (MOOCs)

1. Compiler Design, IIT Madras by PROF. RUPESH NASRE https://nptel.ac.in/courses/106106237
2. Principles of Compiler Design, IISc Bangalore by Prof. Y.N. Srikanth https://nptel.ac.in/courses/106108113
3. NOC:Compiler Design, IIT Kharagpur by Prof. Santanu Chattopadhyay https://nptel.ac.in/courses/106105190

This course can be opted as an elective by the students of following subjects: **MCA**

Suggested equivalent online courses (MOOCs) for credit transfer: N.A.

APPENDIX-II

Guidelines for Research Project/Dissertation

Guidelines for preparing Research Project/Dissertation is available at link:

http://uprtou.ac.in/upload_pdf/01_02_2023_Guidelines_fo_Project_Lit_Survey_Dissertation.pdf