KANNUR UNIVERSITY

(Abstract)

M Sc. Computer Science Programme in the Department of Information Technology, Mangattuparamba Campus - Revised Scheme and Syllabus(I st Semester only)- Approved- Implemented w.e.f. 2023 admission--Orders issued

ACADEMIC C SECTION

ACAD C/ACAD C1/23405/2023 Dated: 23.11.2023

Read:-1. U. O. No. ACAD C/ACAD C3/22373/2019 dtd.12.09.2023

2. Circular No. dated ACAD C/ACAD C3/22373/2019 dated 12/09/2023

3. Email dated 09.11.2023 from the Head, Department of Information Technology, Mangattuparamba Campus.

4.Minutes of the meeting of the Department Council held on 04.10.2023

ORDER

1.The revised Regulations for Post Graduate Programmes under Choice Based Credit and Semester System in the University Teaching Departments/Schools were implemented w.e.f. 2023 admissions vide paper read(1) above.

2. As per paper read (2) above, Heads of all Teaching Departments were requested to submit the revised Syllabus in accordance with the approved regulations along with a copy of the Department Council Minutes

3. As per paper read (3) above, the Head, Department of Information Technology, Mangattuparamba Campus submitted the Scheme & Syllabus (I^{st} Semester only) of M Sc. Computer Science Programme to be implemented in the University Teaching Department w. e. f. 2023 admissions.

4. Department Council vide the paper read (4) above approved the aforementioned Scheme & Syllabus of M Sc. Computer Science Programme to be implemented in the Dept. of Information Technology of the University w. e. f. 2023 admission.

5.The Vice Chancellor after considering the matter in detail, and in exercise of the powers of the Academic Council conferred under section 11(1), Chapter III of Kannur University Act 1996, approved the **Scheme & Syllabus (I st Semester only) of M Sc. Computer Science Programme and accorded sanction to implement the same in the Department of Information Technology, Mangattuparamba Campus of the University w.e.f. 2023 admissions, subject to report to the Academic Council.**

6.The Scheme & Syllabus (I st Semester Only) of M.Sc. Computer Science Programme, under Choice Based Credit Semester System implemented in the Department of Information Technology, Mangattuparamba Campus w. e. f. 2023 admission, is appended and uploaded in the University Web Site.(www.kannuruniversity.ac.in)

7.Orders are issued accordingly.

Sd/-

Narayanadas K
DEPUTY REGISTRAR (ACAD)
For REGISTRAR

To: 1.The Head, Department of Information Technology, Mangattuparamba Campus.
2. Convener, Curriculum Committee.

Copy To: 1. The Examination branch (through PA to CE)
2. PS to VC/ PA to PVC/PA to R
3. DR/AR1/AR II (Acad), EXCI, EP IV
4. Web Manager (for uploading in the website)
5. Computer Programmer
6.SF/DF/FC

Forwarded / By Order

SECTION OFFICER

KANNUR UNIVERSITY
* THAVAKKARA *
CIVIL STATION P.O.
KANNUR
Pin-670 002

DEPARTMENT OF INFORMATION TECHNOLOGY

KANNUR UNIVERSITY

DEGREE OF

MSc. COMPUTER SCIENCE

(CHOICE BASED CREDIT AND SEMESTER SYSTEM)

SYLLABUS

(FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2023 – 24 ONWARDS)

REGULATIONS FOR THE DEGREE OF

MSc. COMPUTER SCIENCE

FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2023 – 24 ONWARDS

PROGRAMME SPECIFIC OUTCOMES

SL #	Outcome
PSO1	Familiar with the entrenched concepts of Computer Science and Applications.
PSO2	Attain skills to design Algorithms and Programs.
PSO3	Design, build up, put into practice and test software systems to meet the given specifications.
PSO4	Enhance the knowledge about emerging topics in Computer Science
PSO5	Attain prerequisite skills to address research problems in computer science.
PSO6	Furnish the candidate to do the real time jobs linked with Information Technology and Computer Application.

1. ADMISSION

Admission to the MSc Computer Science programme will be carried out as per the Kannur University PG regulations – 2023.

Eligibility for Admission:

Basic qualifications: A pass in BCA / Bachelor Degree in Computer Science / Mathematics / Physics/ Statistics / Chemistry / Engineering or equivalent degree.

2. PROGRAMME STRUCTURE

Duration of the MSc Computer Science programme shall be 2 years, divided into 4 semesters. Each semester shall have 18 weeks. The minimum duration for the completion of the MSc Computer Science programme is four (4) semesters from the date of registration. The maximum period of completion is eight semesters (4 years) from the date of registration. Every student have to opt the various categories of course [Discipline Specific Core (DSC), Discipline Specific Electives (DSE), Value Added Courses (VAC), Skill Enhancement Courses (SEC), Ability Enhancement Courses (AEC), Multi-Disciplinary Courses (MDC), MOOC and Inter Disciplinary Courses (IDC)] as per Kannur University PG Regulations - 2023.

For MSc Computer Science programme, all students have to take a MOOC in third semester. Credit earned from the MOOC will be over and above the minimum credit required for completion of MSc Computer Science Degree (Refer Kannur University PG Regulations - 2023).

For the MSc Computer Science programme, a minor project work with 2 credits (offered in the third semester) is opted by the department.

The minimum credits required for the successful completion of the MSc Computer Science programme will be 84. Every student should earn a minimum of 8 credits offered by the other departments.

For all core courses an additional module (called as Module X) is added. Portions of this module need not be discussed by the faculty during the designated classroom hours for the given course. However components of continuous evaluation such as assignment / seminar / viva may be based on the contents of Module X. Module X has to be considered as additional reading and self-exploration by the students. For end semester examinations, contents of Module X will not be considered.

MOOC

Kannur University PG Regulations - 2023 will be applicable for MOOC.

For MSc programme, all students have to choose a MOOC in the third semester. The department council will prepare a list of MOOCs from the online courses offered by the NPTEL / any other MOOC course provider approved by the council from time to time at the beginning of the third semester. Students have to choose a MOOC from this list and earn his/her certificate of completion as per the guidelines stipulated by the MOOC provider from time to time. The credit earned by the student for the MOOC will be counted as such without any normalization.

3. EVALUATION

Evaluation (Both CE and ESE) will be conducted as per the Kannur University PG Regulations - 2023 for all theory and practical courses. However the specific guidelines for conducting the evaluation of theory courses, practical courses, Mini project and Major project are mentioned below.

CONTINUOUS EVALUATION FOR THEORY COURSES (DSC, DSE, VAC, IDC, MDC, AEC, SEC)

CE includes assignments, seminars, viva and periodic written examinations. This should be done by the faculty who manages the course.

The weightage of each component under CE for theory courses shall be in the following proportions:

Components	**% of Weightage**
Test papers (minimum two Continuous Evaluation Tests)	40
Assignments	20
Seminar / Viva	40

Table 1: Weightage of each Component under Continuous Evaluation for Theory Courses

Test Papers: There shall be a minimum of two test papers to be conducted for each course. If more than two test papers are conducted, then two best grades shall be taken for the award of CE grades. The dates of test papers shall be announced well in advance and the result should be displayed in the notice board. Tests such as multiple choice objective type and open text book test (online or offline mode) also can be opted for conducting the test papers.

Assignments: For each course at least one assignment (including practical assignments, if necessary) shall be assigned to the students. The mode of submission and assessment of the assignments shall be decided by the faculty concerned. Assignment works can be conducted either offline/ online mode (as per the decision taken by the faculty concerned).

Viva: Faculty concerned can assign topics for comprehension (based on any portions in the syllabus) and ask the students to appear for individual viva sessions as per a declared schedule. Viva sessions can be conducted either in online or offline mode (as per the decision taken by the faculty concerned).

Seminar: Faculty concerned can assign topics for comprehension (based on the course concerned) and ask the students to prepare seminars based on the topics assigned to them. Each student has to prepare the seminar content and present it. Mode of preparation, submission, and presentation can be specified by the faculty member concerned. Seminar sessions and contents can be submitted and conducted either in online or offline mode (as per the decision taken by the faculty concerned).

Technology Specific Electives (Elective V)

In the fourth semester an Elective course – Technology Specific Elective (Elective V) are meant to foster the students with tools and technologies that they need to know and make use in the design and development of software applications. The department council will prepare the list of elective courses to be offered for Elective V at the end of every third semester. Seminar / Report / Case study implementation report of the specific technology mentioned in the elective should be submitted by each student for the evaluation. The mode of evaluation of this course shall be based on the presentation, report and viva. Both CE and ESE for this course will be conducted by the Department.

Continuous Evaluation For Practical Courses

The components of CE for practical courses are as follows:

Components	% of Weightage
Lab test (minimum 1)	40
Completion of the list of Lab assignments prescribed by the faculty	20
Periodical assessment of assignment in the Lab	40

Table 2: Weightage of each Component under Continuous Evaluation for Practical Courses

Evaluation For Minor Project Work

The components of CE and ESE for minor project work are as follows:

Components	% of Weightage
Understanding of the problem / Concepts	20
Adhering to methodology	15
Quality of presentation and demonstration	15
Quantum of work / effort	25
Organization and content of Project report	5
Viva based on Project	20

Table 3: Weightage of each Component for CE/ ESE for Minor Project Work

CE and ESE of the minor project work shall be done by a departmental committee constituted by the HOD. The committee should consist of a minimum of two faculty members, including the guide. Phases of evaluation and evaluation criteria for each phase shall be framed by the departmental committee.

PROJECT WORK

Project Work offered in the fourth semester currently has 8 credits. Project work has to be undertaken by all students. The project can be software development following all or some of the software development lifecycle or an R & D project. The hours allotted for project work may be clustered into a single slot so that students can do their work at a centre or location for a continuous period of time. The project work should be carried out in the department / Institution / industry / R & D organization of national repute. Project work shall be carried out under the supervision of a faculty member. If the student wishes to undertake his / her project outside the campus, then a co-guide shall be selected from the organization concerned. If the project work is of interdisciplinary nature, a co-guide shall be taken from the other department concerned. Every student should do the project individually and no grouping is allowed. The candidates are required to get the approval of the project synopsis from the supervisor in the department before the commencement of the project. A co-guide should be an expert in the area in which the student has chosen the project. At the end of the semester the candidate shall submit the project report (two bound copies and one soft copy) duly approved by the guide and co-guide for end semester evaluation. The project report shall be prepared according to the guidelines appended along with these regulations / guidelines. Students have to submit the copies of the reports that are approved by the project supervisor(s) before the last date fixed by the department.

The end semester evaluation of the project work shall be done by a board of at least two examiners, in which one should be an external expert. For the evaluation of the project work, the candidate must present the work before the board of examiners which will be followed by a Viva-Voce. The end semester evaluation of the project will be based on the project report, the presentation of the project work undertaken by the student and Viva-Voce.

The weightages for CE and ESE of the project also shall be in the ratio 40:60.

EVALUATION OF PROJECT WORK

CE of the project work shall be done by a departmental committee constituted by the HOD. The committee should consist of a minimum of two faculty members, including the guide.

The assessment is based on presentation, interim report and viva voce. Each internal presentation shall be evaluated based on the following components:

Components	% of Weightage
Understanding of the problem /Concepts	20
Adhering to methodology	15
Quality of presentation and demonstration	15
Quantum of work / effort	25
Organization and content of Project report	5
Viva based on Project	20

Table 4: Components for Continuous Evaluation and the Corresponding Weightage (for Project Work)

End Semester Evaluation (ESE): A board of two examiners appointed by the university shall conduct ESE. The evaluation shall be based on the report, presentation of the work, demonstration of the work and a detailed viva voce based on the work carried out. A candidate will not be permitted to attend the project evaluation without project reports that are duly certified by the guide and HOD. Also, a project will be evaluated only if the candidate attends the ESE presentation and Viva voce on the scheduled date and time. A board shall evaluate a maximum of 10 candidates in a day. The ESE shall consist of the following components:

COMPONENTS	% Weightage
Understanding of the problem/requirements/	15

concepts related to the project	
Adhering to methodology (Software engineering phases or research methodology) and the candidates understanding of the components of methodology	15
Quality of Modelling of the problem and solution/ database design / form design / reports / testing (For research projects - relevance / novelty of the work(s) / use of data/ proposal of new models /analysis of algorithms/ comparison and analysis of results / findings)	20
Quality of presentation / demonstration	15
Quantum of work / effort - assessed through the content of report, presentation and viva	25
Organization and content of report	10

Table 5: Components for ESE with the Corresponding Weightage (for Project Work)

Guideline for Preparing Project Report (Both Minor Project Work and Project Work)

i) Arrangement of contents:

The sequence in which the project report material should be arranged and bound should be as follows:

1) Cover Page & Title Page

2) Plagiarism Report

3) Bonafide Certificate

4) Abstract

5) Table of Contents

6) List of Tables

7) List of Figures

8) List of Symbols, Abbreviations and Nomenclature

9) Chapters

10) Conclusion

11) Publications based on the project work (if any)11) Appendices

12) References

The chapters may be broadly divided into 3 parts: (i) introductory chapter, (ii) chapters developing the main theme of the project work, (iii) implementation details (if any) and conclusion. The main text will be divided into several chapters and each chapter may be further divided into several divisions and subdivisions. Each chapter should be given an appropriate title.

Tables and figures in a chapter should be placed in the immediate vicinity of the reference where they are cited. The tables and figures shall be introduced at appropriate places.

Footnotes should be used sparingly. They should be typed single space and placed directly underneath in the very same page, which refers to the material they annotate.

ii) Page Dimension and Binding Specifications:

The dimension of the project report should be in A4 size. The project report should be bound using a flexible cover of the thick white art paper. The cover should be printed in black letters and the text for printing should be identical.

All the project reports submitted by the students should be plagiarism checked using software and the plagiarism report generated by the software should be verified and signed by the HOD.

MSc Computer Science

LEARNING OUTCOME BASED CURRICULUM FRAME WORK &PROGRAMME STRUCTURE

Graduate Attributes

GRADUATE ATTRIBUTE 1 - SCHOLARSHIP

KANNUR UNIVERSITY graduates will be able inquire critically into their area of study, while being aware of changing state of knowledge both in their own chosen discipline as well as related disciplines

KANNUR UNIVERSITY graduates will have the ability to actively engage in the generation of innovative and relevant knowledge and understanding through inquiry, critique and synthesis going beyond their discipline of specialization.

GRADUATE ATTRIBUTE 2 - CRITICAL CITIZENSHIP AND THE SOCIAL GOOD

KANNUR UNIVERSITY graduates will be engaged, committed and accountable agents of social good. They must aspire to contribute to social justice and environmental sustainability, appreciative of the complexity of historical contexts and societal conditions through their roles as professionals and members of local and global communities.

KANNUR UNIVERSITY graduates will be committed to furthering gender and social equality and empathetically engage with all forms of difference including, conflicting intellectual traditions, religious and cultural practices, language, region and nationality.

GRADUATE ATTRIBUTE 3 - LIFELONG LEARNING

KANNUR UNIVERSITY graduates will be Lifelong Learners, committed to and capable of continuous collaborative and individual learning and critical reflection for the purpose of furthering their understanding of the world and their place in it.

Programme Outcomes (PO)

SL #	Outcome
PO1	**Critical Thinking**: Take informed actions after identifying the assumptions that frame our thinking and actions, checking out the degree to which these assumptions are accurate and valid, and looking at our ideas and decisions (intellectual, organizational, and personal) from different perspectives
PO2	**Problem Solving**: Identify, formulate, conduct investigations, and find solutions to problems based on in-depth knowledge of relevant domains
PO3	**Communication**: Speak, read, write and listen clearly in person and through electronic media in English/language of the discipline, and make meaning of the world by connecting people, ideas, books, media and technology.
PO4	**Responsible Citizenship**: Demonstrate empathetic social concern, and the ability to act with an informed awareness of issues
PO5	**Ethics**: Recognize different value systems including your own, understand the moral dimensions of your decisions, and accept responsibility for them.
PO6	**Self-directed and Life-long Learning**: Acquire the ability to engage in independent and life-long learning in the broadest context socio-technological changes
PO7	**Environmental Sustainability and Global Perspective:** - Develop an understanding of global standards to foster legal environment. Learn and practice to critically analyse the legal issues from local, national and international concerns

Programme Specific Outcomes (PSO)

SL #	Outcome
PSO1	Familiar with the entrenched concepts of Computer Science and Applications.
PSO2	Attain skills to design Algorithms and Programs.
PSO3	Design, build up, put into practice and test software systems to meet the given specifications.
PSO4	Enhance the knowledge about emerging topics in Computer Science
PSO5	Attain prerequisite skills to address research problems in computer science.
PSO6	Furnish the candidate to do the real time jobs linked with Information Technology and Computer Application.

Mapping of PSOs to POs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
PSO1	✓	✓	✓			✓	✓
PSO2	✓	✓				✓	✓
PSO3	✓	✓	✓	✓	✓	✓	✓
PSO4	✓	✓	✓	✓	✓	✓	✓
PSO5	✓	✓	✓	✓	✓	✓	✓
PSO6	✓	✓	✓	✓	✓	✓	✓

MSc. Computer Science

Programme Structure

Course Structure

Distribution of Credits for the MSc. Computer Science Programmes with effect from 2023- 24 Onwards									
	1	2	3	4	5	6	7	8	Total Credits
	Discipline Specific		Electives						
Sem	Core (DSC)	Elective (DSE)	Interdisciplinary / Multidisciplinary / Open	Ability Enhancement Course (AEC) 2 Credits	Skill Enhancement Course (SEC) 2 Credits	Value Added Course / MOOC 2 Credits	Internship / Field Visit / Minor Project / Institutional Industrial Visit 2 Credits	Dissertation / Major Project	

1	MSCSC01DSC01(3C)
	MSCSC01DSC02(4C)
	MSCSC01DSC03(3C)
	MSCSC01DSC04(4C)
	MSCSC01DSC05(2C)
	MSCSC01DSC06(2C)
		MSCSC01DSE01
		MSCSC01DSE02
		MSCSC01DSE03							

	18 Credits	3 Credits each							21

2	MSCSC02DSC07(3C)
	MSCSC02DSC08(3C)
	MSCSC02DSC09(3C)
	MSCSC02DSC10(3C)
	MSCSC02DSC11(3C)
	MSCSC02DSC12(2C)
	MSCSC02DSC13(2C)
				MSCS02AECXX (2C)	MSCS02SECXX (2C)				

	19 Credits			4 Credits from Any of These					23

3	MSCSC03DSC14(4C)
	MSCSC03DSC15(4C)
	MSCSC03DSC16(3C)
	MSCSC03DSC17(2C)
		MSCSC03DSE04
		MSCSC03DSE05
		MSCSC03DSE06
		MSCSC03DSE07
		MSCSC03DSE08
		MSCSC03DSE09
			MSCSC03MDC01 to 7 (Offered for Other Department			MSCSC03VACXX	MSCSC03DSC18		

	13Credits	3 Credits Each	4 Credits Each			2 Credits*	2 Credits		22
4		Elective III Elective IV (MSCSC04DSE10 to 25) (4C) Elective V MSCSC04DSE26(2C)						MSCSC04DSC19 (8C)	
		10 Credits from Any of These						8 Credits	18

*MOOC Credit will not be counted for CGPA. However, it is compulsory. Credit earned by the students for the MOOC will be entered as provided by the MOOC provider.

LEGEND	
Item	**Description**
C	Credits
E	External Component (Marks)
ESA	End Semester Evaluation
CE	Continuous Evaluation
P	Practical Hours
T	Total
Tt	Tutorial
CS	Case Study
VAC	Value Added Course
SEC	Skill Enhancement Course
AEC	Ability Enhancement Course
TEC	Technology Enhancement Course
DSC	Department Specific Core
DSE	Department Specific Elective
MDC	Multidisciplinary Elective Course

Semester I

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
1.1	MSCSC01DSC01	Mathematical Foundations for Computer Science	3	3	0	1	60	40	100
1.2	MSCSC01DSC02	System Software and Operating Systems	4	4	0	1	60	40	100
1.3	MSCSC01DSC03	Time Series Analysis and Forecasting	3	3	0	1	60	40	100
1.4	MSCSC01DSC04	Database Management System	4	4	0	1	60	40	100
1.5	MSCSC01DSEXX	Elective I-DSE (POOL A)	3	3	0	1	60	40	100
1.6	MSCSC01DSC05	Lab I: (i)DBMS (ii) SSOS	2	0	4	0	60	40	100
1.7	MSCSC01DSC06	Lab II: Principles of Programming and Numerical Methods	2	0	4	0	60	40	100
		Total	**21**	**17**	**8**	**5**			

S1 - List of Electives for DSE (POOL A)		
No	**Course Code**	**Course Name**
1	MSCSC01DSE01	Principles of Programming and Numerical Methods using Python
2	MSCSC01DSE02	Principles of Programming and Numerical Methods using C
3	MSCSC01DSE03	Principles of Programming and Numerical Methods using C++

Semester II

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
2.1	MSCSC02DSC07	Algorithms and Data Structure	3	3	0	1	60	40	100
2.2	MSCSC02DSC08	Digital Signal Processing	3	3	0	1	60	40	100
2.3	MSCSC02DSC09	Artificial Intelligence	3	3	0	1	60	40	100
2.4	MSCSC02DSC10	Theory of Computation	3	3	0	1	60	40	100
2.5	MSCSC02DSC11	Computer Network and Linux Administration	3	3	0	1	60	40	100
2.6	MSCSC02DSC12	Lab III: i) Algorithms and Data Structure	2	0	4	0	60	40	100
2.7	MSCSC02DSC13	Lab IV: (a)Digital Signal Processing (b)Time Series Analysis and Forecasting	2	0	4	0	60	40	100
2.8	Offered by other departments	SEC/AEC	2	2	0	0	60	40	100
2.9	Offered by other departments	SEC/AEC	2	2	0	0	60	40	100
		Total	**23**	**19**	**8**	**5**			

Semester III

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
3.1	MSCSC03DSC14	Machine Learning Techniques	4	4	0	1	60	40	100
3.2	MSCSC03DSC15	Computer Graphics and Image Processing	4	4	0	1	60	40	100
3.3	MSCSC03DSC16	Quantum Computing	3	3	0	1	60	40	100
3.4	Offered by other departments	IDC/MDC	4	4	0	0	60	40	100
3.5	MSCSC03DSEXX	Elective II -DSE (POOL B)	3	3	0	0	60	40	100
3.6	MSCSC03DSC17	Lab V: (i)Machine Learning (ii) CG and IP	2	0	4	1	60	40	100
3.7	MSCSC03DSC18	Mini Project	2	0	3	1	60	40	100
3.8	MSCSC03VAC01	MOOC	2*	0	0	0			**
		Total	**22**	**18**	**7**	**5**			

*MOOC Credit will not be counted for CGPA. However it is compulsory.

** Marks earned by the students for the MOOC will be entered as provided by the MOOC provider.

S3 - List of Electives for DSE (POOL B)

No	Course Code	Course Name
1	MSCSC03DSE04	Foundations of Data Science
2	MSCSC03DSE05	Foundations of Natural Language Processing
3	MSCSC03DSE06	Speech Audio and Video Forensics
4	MSCSC03DSE07	Internet of Things
5	MSCSC03DSE08	Pattern Recognition
6	MSCSC03DSE09	Computer Vision

Multi-disciplinary Course (offered for other Departments)

MDC - 3rd Semester

No	Course Code	Course Name	C	T	P	Tt
1	MSCSC03MDC01	Design and Analysis of Algorithms	4	4	0	0
2	MSCSC03MDC02	Principle of Programming and Numerical Methods	4	4	0	0
3	MSCSC03MDC03	Java Programming	4	4	0	0
4	MSCSC03MDC04	Machine Learning	4	4	0	0
5	MSCSC03MDC05	Foundations in Data science	4	4	0	0
6	MSCSC03MDC06	Digital Signal Processing	4	4	0	0
7	MSCSC03MDC07	Quantum Computing and Information Theory	4	4	0	0
	Total		4	4	0	0

Semester IV

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
4.1	MSCSC04DSEXX	Elective III-DSE (POOL C)	4	4	0	0	60	40	100
4.2	MSCSC04DSEXX	Elective IV- DSE(POOL C)	4	4	0	0	60	40	100
4.3	MSCSC04DSE26	Elective V*	2	2	0	0	60	40	100
4.4	MSCSC04DSC19	Project	8	0	15	5	60	40	100
		Total	18	10	15	5			

*Elective V is meant to foster the students with tools and technologies that they need to know and make use in the design and development of software applications. Seminar/Report/ Case study implementation report of a specific technology should be submitted by each student for the evaluation. The mode of evaluation of this course shall be based on the presentation, report and viva.

S4 - List of Discipline Specific Electives (DSE III/DSE IV) (POOL C)

No	Course Code	Type of Elective	Course Name
1	MSCSC04DSE10	DSE	Software Engineering
2	MSCSC04DSE11	DSE	Cyber Physical Systems
3	MSCSC04DSE12	DSE	Digital Forensics
4	MSCSC04DSE13	DSE	Block chain and Crypto currency technologies
5	MSCSC04DSE14	DSE	Grid and Cloud Computing
6	MSCSC04DSE15	DSE	Embedded System
7	MSCSC04DSE16	DSE	High Performance Computing
8	MSCSC04DSE17	DSE	Data and Information Visualization
9	MSCSC04DSE18	DSE	Information Retrieval System
10	MSCSC04DSE19	DSE	Biometric Image Processing
11	MSCSC04DSE20	DSE	Java Programming
12	MSCSC04DSE21	DSE	Nature Inspired Computing
13	MSCSC04DSE22	DSE	Big Data Analytics
14	MSCSC04DSE23	DSE	Astronomical Image Processing
15	MSCSC04DSE24	DSE	Geographic Information System
16	MSCSC04DSE25	DSE	Operation Research

Semester I

CORE COURSE

MSCSC01DSC01 MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course deals with mathematical concepts like elementary discrete mathematics, probability & statistics and linear algebra.

Course Objectives:

- Impart knowledge on mathematical logic.
- Give basic idea set theory, relations and functions and their problem solving.
- Familiarize measures of central tendency and measures of dispersion.
- Impart knowledge on probability and its distributions.
- Familiarize matrices and its operations, vector space and Eigen vectors.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Acquire knowledge about mathematical logic, set theory and relations
CO2	Use functions, partial ordering and counting techniques to solve problems.
CO3	Understand measures of central tendency, measures of dispersion, probability and its distributions.
CO4	Basic understanding of linear algebra

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓	✓		✓	✓
CO2	✓	✓			✓	✓
CO3	✓			✓	✓	✓
CO4	✓	✓	✓	✓	✓	✓

COURSE CONTENTS

Module 1: Mathematical logic: Propositional and Predicate Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Normal Forms

Set and relations: Set Operations, Properties of Relations, Representing relations-matrices & digraphs. Closure of Relations, Composition of relations, Equivalence Relations.

Module 2: Functions: Types of Functions, Composition of Functions and Inverse Functions, Some important functions- floor & ceiling, Recursive functions

Partitions- Partial Ordering, Hasse Diagram, Lattice –Types, Properties, Basics of Counting, Pigeonhole Principle, Permutations and Combinations, Inclusion- Exclusion Principle.

Module 3: Measures of Central Tendency, Measures of Dispersion, Coefficient of Variation, Covariance.

Probability – Random experiment, Sample point, Sample space, Events, Algebra of events, Statistical regularity, Frequency and Classical definitions, Axiomatic approach to probability, Probability Space and probability measure, Addition theorem, Conditional probability, Multiplication theorem, Independence of events, Bayes' theorem and applications.

Discrete Distributions – One point, two point distributions, Uniform, Point binomial, Poisson, Continuous Distributions – Normal, Exponential

Module 4: Matrices and determinants: matrix, types of matrices, operations on matrices, transpose of a matrix, Determinants-properties of determinants- inverse of a matrix- Rank of a Matrix, Trace of a Matrix. Solving Linear Equations using Matrices – Matrix solution, Gauss Elimination Method

Vector Space, Subspace, Linear Dependence and Independence, Basis and Dimension, Linear Transformations, Matrices Related to Linear Transformations, Eigen values and Eigenvectors.

Module X (For Additional Reading and Comprehension by the Students):

Mathematical Induction, Recurrence Relation, Generating function, Group Theory: Groups, Subgroups

Discrete Distributions– Geometric, Hyper geometric and Negative binomial distributions

Continuous Distribution - Rectangular, Beta, Gamma, log normal distribution. Consistency of a linear system, Diagonalization of a matrix, Diagonalization of a symmetric matrix.

Core Compulsory Readings

1. Kenneth H. Rosen, Discrete Mathematics and Applications, TMH 2003
2. Elementary Linear Algebra - Devi Prasad (Narosa Pub. House, 2006)
3. Fundamentals of Mathematical Statistics - S. C.Gupta &V.K.Kapoor (Sulthan Chand & Sons)

Core Suggested Readings

1. Discrete Mathematics and Its Applications with Combinatorics and Graph Theory, Kamala Krithivasan, McGraw Hill Education, 2011 (Seventh Edition).
2. J.P.Tremblay and R Manohar, Discrete Mathematical Structures with Applications to Computer Science, TMH 2001
3. Discrete Mathematics, N Ch S N Iyengar, V M Chandrasekharan, KA Venkatesh, PS Arunachalam, Vikas Publishing, 2003.
4. Introduction to Probability and Statistics for Engineers and Scientists- S.M. Ross (Elsever)
5. Linear Algebra - A Geometric Approach - S. Kumaresan (Prentice Hall India)

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar…..	40%

Sample Questions to test Outcomes.

1. Define tautology and contradiction.
2. Show that for any two sets $A-(A \cap B) = A-B$
3. Explain Equivalence relation.
4. Let R be a partial ordering of the set of all devisors of 64. Construct the hasse diagram for it. Find the meet and join.
5. Three dice are rolled together. What is the probability of getting at least one '4'?
6. Use Gauss elimination to solve the following system of linear equations.

$$2X + Y + Z = 10$$

$$3X + 2Y + 3Z = 18$$

$$X + 4Y + 9Z = 16$$

CORE COURSE

MSCSC01DSC02 SYSTEM SOFTWARE AND OPERATING SYSTEMS

Credit			**Teaching Hours**			**Assessment Weightage (%)**		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4/1	0	5	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course is to provide students with basic knowledge of system software. This course will cover assemblers, linkers, loaders and compilers. Particular emphasis will be given to major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), file systems, I/O systems and mass storage structure.

Course Objectives:

- To know the design and implementation of assemblers, macro processor, linker, loader and compiler.
- To explain the main components of OS and their working.
- To familiarize the operations performed by OS as a resource Manager.
- To impart various scheduling policies of OS
- To teach the different memory management techniques.
- To explain file system, mass storage structure and input/output management.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Acquire knowledge about Assembler, Linkers and Loaders.
CO2	Understand Process scheduling, process synchronization and methods to handle deadlocks.
CO3	Understand Memory Management and file management techniques.
CO4	Understand I/O systems, mass storage structure and different disk scheduling algorithms.

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓	✓		✓	✓
CO2	✓	✓	✓		✓	✓
CO3	✓	✓	✓	✓	✓	✓
CO4	✓	✓		✓	✓	✓

COURSE CONTENTS

Module 1: Assemblers: Elements of Assembly Language Programming, Overview of Assembly Process, Design of Two pass Assembler, Macros and Macro Processors, Macro definition, call and expansion, Nested Macro calls, Design of Macro preprocessor. Linkers: Linking and Relocation concepts, Design of linkers, Self relocating programs. Loaders: introduction to loaders - functions of loaders- Compilers: Introduction to compilers -Different Phases-Lexical Analysis- role of the lexical analyzer, input buffering, specification of tokens, Recognition of tokens, lexical Analyzer generators, Lex.

Module 2: Introduction to Operating systems: Different types of Operating system, Overview of Operating systems, Operating system structures, Process management - Processes, Process Scheduling, Inter Process communication - Communication in client server systems, Threads - Processes Vs Threads, Types of threads, Multicore and Multithreading. CPU Scheduling - Scheduling algorithms. Process synchronization: Critical section Problem, Mutual Exclusion, Requirements, Semaphores, Producer Consumer Problem, Readers Writers Problem, Deadlock :Prevention, Detection and Recovery.

Module 3: Memory Management- Swapping, Contiguous memory allocation, Paging, Segmentation, Segmentation with paging. Virtual memory- Demand paging, processes creation, page replacement, allocation of frames, thrashing. File system interface and Implementation - File concepts, access methods, directory structure, File system implementation, Directory implementation, Allocation methods.

Module 4: I / O Systems **-** I / O hardware, Application I/O interface, Kernel I / O subsystem, Transforming I / O to hardware operations, STREAMS, Performances. Mass storage structure - Disk structure, Disk scheduling, Disk management, Swap space managements, RAID structure, Disk attachments, Stable storage implementation, Tertiary storage structure.

Module X (For Additional Reading and Comprehension by the Students): Macros-Advanced Macro facilities, Linker-Linking for over-lays, CPU Scheduling- Multiple Processor Scheduling, Algorithm Evaluation- Advanced CPU scheduling. Process synchronization: Monitors, File system interface and Implementation- File system mounting, File sharing, Protection, File system structure, Free space managements, Efficiency and performance, Recovery, Log- structured file system.

Core Compulsory Readings

1. D.M. Dhamdhere, Systems Programming and Operating Systems, TMH, 2003.
2. Silberschatz, A., Galvin, P.B. & Gagne, G. Operating System Concepts, 9th Ed. John Wiley & Sons- India.

Core Suggested Readings

1. Dhamdhere, D. M. Operating Systems, 2nd Ed. The McGraw - Hill Companies.
2. Ditel, Deital and Choffness, Operating Systems, Pearson, 3rdEdn
3. William Stallings, Operating Systems, Internals and Design Principles, 7th Edition Pearson,
4. Sibsankar Haldar ,Alex a Aravind, Operating Systems, Pearson Education India, Second impression.
5. Andrew S.Tanenbaum , Albert S.Woodhull, The Minix Book- Operating Systems Design and Implementation, 3rd Edition Pearson(2016).

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Print

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%

Assignment	20%
Seminar.....	40%

Sample Questions to test Outcomes.

1. Design a two-pass assembler with its necessary phases
2. With an example explain FCFS, SJF and Round Robin CPU Scheduling Algorithms
3. Describe Producer Consumer Problem
4. Illustrate banker's algorithm to avoid deadlocks
5. Describe Paging mechanism with examples
6. Implement any three-page replacement algorithms for the reference string given as follows and find the number of page faults
7. Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.
8. Explain various File Access methods involved with OS
9. Explain with a diagram a typical PC Bus structure with I/O Devices
10. Apply and illustrate SSTF, FCFS and SCAN disk scheduling algorithms with a request queue of 98, 183, 37, 122, 14, 124, 65, 67 with Head pointer at 53

CORE COURSE

MSCSC01DSC03 TIME SERIES ANALYSIS AND FORECASTING

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

Time Series Analysis and Forecasting is a practical course that teaches students the essential concepts and techniques for analysing and predicting patterns in time-dependent data. Students will learn how to pre-process and visualize time series data, identify trends and seasonality, and select appropriate models for analysis. The course emphasizes hands-on experience with popular statistical software packages and real-world case studies to reinforce learning. By the end of the course, students will be equipped with the skills to effectively analyse time series data and make accurate forecasts, enabling them to make informed decisions in diverse fields such as finance, economics, and marketing.

Course Objectives:

- Understand the fundamental concepts and characteristics of time series data, including autocorrelation, stationarity, trends, and seasonality.
- Learn various techniques for preprocessing and visualizing time series data to uncover patterns and insights.
- Develop proficiency in selecting and implementing appropriate time series models, such as ARIMA, exponential smoothing, and state space models.
- Gain hands-on experience in forecasting by applying different techniques and evaluating the performance of forecasting models.
- Apply time series analysis and forecasting skills to real-world case studies and projects, enabling students to make data-driven decisions and predictions in practical scenarios.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Develop comprehensive understanding of probability concept.
CO2	Perform Data Pre-processing Techniques, Derive the properties of ARIMA and state-space models.
CO3	Choose an appropriate ARIMA model for a given set of data and fit the model using an appropriate package.
CO4	Compute forecasts for a variety of linear methods and models

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓		✓	✓	✓
CO2	✓	✓		✓	✓	✓
CO3		✓	✓	✓	✓	✓
CO4		✓	✓	✓	✓	✓

COURSE CONTENTS

Module 1: Random variable – Continuous, Discrete, Identically Independent Distribution (IID), Introduction to probability, Conditional probability, Bayes theorem, Probability Distribution, Probability Density function, Expectation value, Moment of probability density function, variance, covariance, standard deviation, auto covariance, central limit theorem, correlation, autocorrelation, partial autocorrelation, random process, random walk, Difference between time series and regression, Difference between probability and likelihood

Module 2: Fourth paradigm of science, Introduction to time series data, characteristics of time series data, Understanding the components of a time series: trend, seasonality, cyclicality, and residual, Stationary and non-stationary Time Series, Vector Valued and Multidimensional Series._Introduction to Data Acquisition and Pre-processing, Data Cleaning – outlier detection and treatment, handling missing data, Smoothing, Detrending, Data Normalization, Data Transformation, Data Integration, Feature Selection, Handling imbalanced data

Module 3: Exploratory data analysis, Explore patterns and trends in time series data, Seasonality analysis and detection methods, Autocorrelation and partial autocorrelation analysis. Trend analysis and trend removal techniques, Seasonal decomposition and visualization of components. Classical Regression in the Time Series Context

Module 4: Introduction to ARMA models, Model identification: selecting appropriate orders of ARIMA models, Parameter estimation and model fitting, Model diagnostics and evaluation: residual analysis, Introduction to SARIMA model. Time series forecasting using ARIMA and SARIMA models.

Module X (For Additional Reading and Comprehension by the Students): Recurrent Neural Networks (RNNs) - Introduction to RNNs, Understanding the recurrent layer,Long Short-Term Memory (LSTM) networks, Gated Recurrent Unit (GRU) networks, Advantagesand use cases

Core Compulsory Readings

1. Rob J Hyndman (2014), Forecasting: Principle & Practice, University of Western Australia
2. R. H. Shumway and D. S. Sto_er (2017), Time Series Analysis and Its Applications (With R Examples, fourth Edition).Springer, New York.

Core Suggested Readings

1. Nonlinear time series analysis .Ruey S. Tsay and Ronngchen. Wiley 2019
2. Enders W. Applied Econometric Time Series. John Wiley & Sons, Inc., 1995.
3. Mills, T.C. The Econometric Modelling of Financial Time Series. Cambridge University Press, 1999
4. Andrew C. Harvey. Time Series Models._Harvesterwheatsheaf, 1993.
5. Andrew C. Harvey. The Econometric Analysis of Time Series. Philip Allan, 1990.

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Print

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar…..	40%

Sample Questions to test Outcomes.

1. What are the key characteristics of time series data, and why is it important to understand them in the context of analysis and forecasting?
2. How can you pre-process time series data to handle missing values and outliers effectively?
3. What are the different methods available for visualizing time series data, and how can they help in identifying trends and seasonality?
4. Explain the concept of autocorrelation and its significance in time series analysis.
5. What is stationarity in the context of time series data, and why is it important for modelling and forecasting?
6. Compare and contrast the ARIMA and exponential smoothing models in terms of their assumptions and applicability.
7. How can you evaluate the performance of a forecasting model, and what are some commonly used metrics for this purpose?
8. Describe the process of model selection in time series analysis and the factors to consider when choosing an appropriate model.
9. What are the steps involved in building a forecast using the chosen time series model, and how can you interpret the results?
10. Provide examples of real-world applications of time series analysis and forecasting in different industries or domains.

CORE COURSE

MSCSC01DSC04 DATABASE MANAGEMENT SYSTEM

Credit	Teaching Hours	Assessment Weightage (%)

L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4/1	0	5	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

The Database Management Systems (DBMS) course is designed to provide a comprehensive understanding of the principles and practices of managing databases. Also focuses on the theoretical foundations and practical applications of DBMS, which play a crucial role in modern information systems.

Course Objectives:

- To introduce students to the basic concepts and the principles of Database Management Systems.
- To understand the value of data integrity, security and privacy concepts.
- To explore normalization, transaction management, indexing and concurrency control.
- To provide a hands-on experience in designing, implementation and management of databases.
- To introduce emerging technologies in DBMS field.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the concept of Database Management Systems
CO2	Understand the concept of transaction management with respect to database.
CO3	Compare different types of NoSQL Databases and their applications
CO4	Demonstrate the detailed architecture and performance tune of Document-oriented NoSQL databases.

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓		✓	✓	✓	✓
CO2	✓		✓	✓	✓	✓
CO3			✓	✓	✓	✓
CO4			✓	✓	✓	✓

COURSE CONTENTS

Module 1: Introduction to Database Management Concepts: The ER and EER. Relational Algebra and Relational calculus. Functional dependencies -1st, 2nd, 3rd, 4th, BCNF, 5th Normal form. Sub-queries: correlated sub-queries, join, Exist, Any, All, joined relations. Integrity and security: domain constraints, Referential integrity, assertion, triggers. Views: Introduction to views, data independence, security, updates on views, comparison between tables and views.

Module 2: Transaction processing- desirable properties of transaction. Transactions and Schedules –Characterising Schedules based on Recoverability, Serializability of schedules. Concurrency Control in databases: Locking Techniques-Time stamp ordering, Multi version concurrency Control –Granularity of data items.

Module 3: Overview and History of NoSQL Databases. Definition of the Four Types of NoSQL Databases, The Value of Relational Databases, Getting at Persistent Data, Concurrency, Integration, Impedance Mismatch, The Emergence of NoSQL, Aggregate Data Models; Aggregates, Example of Relations and Aggregates, Consequences of Aggregate Orientation, Application and Integration Databases, Attack of the Clusters.

Module 4: Map-Reduce: Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations, A Two Stage Map-Reduce Example, Incremental Map-Reduce Key-Value Databases, What Is a Key-Value Store, Key-Value Store Features, Consistency, Transactions, Query Features, Structure of Data, Scaling, Suitable Use Cases, Storing Session Information, User Profiles, Preference, Shopping Cart Data, When Not to Use, Relationships among Data, Multioperation Transactions, Query by Data, Operations by Sets.

Module X: (For additional reading and comprehension by the students)

NoSQL Key/Value databases using MongoDB, Document Databases, Document oriented Database Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Complex Transactions Spanning Different Operations, Queries against Varying Aggregate Structure.

Core Compulsory Readings

1. HSilbersehatz, Korth and Sudarshan, Database system concepts, 6th edition MGH 2011
2. Ramakrishnan and Gehrke, Database Management Systems, 3rd Edn, Mc Graw Hill, 2003
3. Elmasri and Navathe, Fundementals of Database systems, 5th Edition, Pearson 2009
4. C.J.Date-A.Kannan, S.Swamynathan "An introduction to Database System" 8th Edition, Pearson education O'Reilly, Practical PostgreSQL Shroff Publishers(SPD) 2002.
5. Sadalage, P. & Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. (1st Ed.). Upper Saddle River, NJ: Pearson Education, Inc. ISBN- 13: 978-0321826626 ISBN-10: 0321826620
6. Redmond, E. & Wilson, J. (2012). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement (1st Ed.). Raleigh, NC: The Pragmatic Programmers, LLC. ISBN-13: 978-1934356920 ISBN-10: 1934356921

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar.....	40%

Sample Questions to test Outcomes

1. Explain the concept of normalization in the context of database design. Why is it important? Give an example.
2. With corresponding examples, explain the difference between a primary key and a foreign key in a relational database. How it can be represented in a table.
3. Discuss the concept of ACID properties in transaction management. How do they ensure data integrity and consistency.

4. Write the advantages and disadvantages of various database indexes.
5. Write an SQL query to retrieve all students who have got more than 75 percentage of mark in the department of science from the student table.
6. Explain the purpose of a trigger in database. Give an example scenario where a trigger can be useful.
7. Discuss how database concurrency control used to ensure the consistency of data in a multi-user environment.
8. Discuss the role of data warehousing in decision support systems. What are the common challenges to be faces while implementing a data warehouse.
9. Compare and contrast distributed database system and centralized database system.
10. What are NoSQL databases? How it is different from traditional databases? Discuss a scenario where NoSQL database is more suitable than the traditional database.

ELECTIVE COURSE

MSCSC01DSE01PRINCIPLES OF PROGRAMMING AND NUMERICAL METHODS USING PYTHON

Credit			**Teaching Hours**			**Assessment Weightage (%)**		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

Numerical computational methods help to solve complex mathematical problems which cannot be solved easily by analytical mathematics by using simple arithmetic operations and which requires the use of an algorithm. The understanding of numerical computing helps to simulate various scientific models. The course will focus on different advanced paradigms and approaches using Python programming.

Course Objectives:

- To introduce basic concepts of python programming language.
- Discuss about Errors and Approximations
- Discuss about the concept of numerical computational methods
- Discuss the numerical integration and differentiation:

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Acquainted with Numerical Methods.
CO2	Obtain the knowledge about Numerical Integration,Differentiationand its applications
CO3	Understanding the basic concepts of Python programming.
CO4	Ripen skill in programming

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓		✓	✓	✓
CO2	✓	✓		✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓	✓

COURSE CONTENTS

Module 1: Introduction to Numerical Methods: Nature of numerical problems; computer based solutions. Errors and Approximations. Nonlinear equations – Bisection Method, Regular- Falsie Method, Newton- Raphson. System of Linear Equations- Gauss elimination, Gauss Jordan elimination, Triangulation method, Iterative method, Jacobi. Case study by writing algorithms.

Module 2: Numerical Integration and Differentiation: Concept of differentiation and Integration. Taylors series and Eulers methods-Simpson's Romberg, Gaussian, Runge-Kutta methods. Case study by writing algorithms.

Module 3: Features of Python, Different Methods to Run Python, Basic Elements (Objects, Expressions, Numerical Types, Strings, Variables), Comments, Indentation in Python, Input and Output in Python, import function, Operators in Python, Branching (if, else, elif), Iteration (while, for), range and enumerate functions, Tuples, Lists, Sets, Dictionaries, Built-in methods of lists, sets and dictionaries, Mutable and Immutable Objects.

Module 4: Functions Definition, Function Calling, Function Arguments (Required, Keyword, Default), Recursion, Modules, Built-in Modules, Creating Modules, File Handling (Opening, Closing, Writing, Reading), Exceptions, Built-in Exceptions (IndexError, OverflowError, ZeroDivisionError, RuntimeError), Exception Handling. Class Definition, Object Creation, Built-in Attribute Methods, Object Oriented Programming Features of Python. Arrays in Python, Numpy Module, ndarray, Creating Arrays (array, zeros, ones, empty, linspace, arrange, random), Two-Dimensional Array, Indexing, Slicing, Iterating, Copying, Splitting, Shape Manipulation (reshape, transpose, resize), Arithmetic Operations on Arrays. Data Visualization in Python matplotlib Module, pyplot, plot(), scatter, bar charts, Formatting, figure(), subplot(), text(), xlabel(), ylabel(), title(), Plotting Simple Mathematical Functions (sin x, x^2).

Core Compulsory Readings

1. Discrete Mathematical Structures with Application to Computer Science-McGraw Hill
2. Introductory Methods of Numerical Analysis – January 2012 PHI, Sastry S.S

Core Suggested Readings

1. Balagurusamy, E., "Numerical Methods", Tata McGraw-Hill, New Delhi, 1999.
2. R.G.Dromey , How to solve it by computer, Pearson education, fifth edition, 2007.
3. Taming Python By Programming, Dr. Jeeva Jose, Khanna Publishing
4. Introduction to Computation and Programming Using Python with Application to Understanding Data - John V. Guttag, PHI (2016)
5. https://www.numpy.org/devdocs/user/quickstart.html
6. https://matplotlib.org/users/pyplot_tutorial.html

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Print

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar…..	40%

Sample Questions to test Outcomes.

1. Write python program to demonstrate Bisection method

2. Write the algorithm of False position method

3. Explain Gaussian elimination method with an example

4. Write the algorithm to implement Gauss-Jordan method

5. Write python program for computing the value of nP_r

6. Explain the concept of differentiation and Integration with examples.

7. Find the real root of the equation $x^3-2x-5 = 0$

8. Find the positive root, between 0 and 1, of the equation $x = e^{-x}$ to a tolerance of 0.05%

9. Find a root, correct to three decimal places and laying between 0 and 0.5, of the equation

$4\ e^{-x}\ \text{Sin}\ x - 1 = 0$

10. Obtain the solution of the following system using the Jacobi iteration method

$2x_1 + x_2 + x_3 = 5$

$3x_1 + 5x_2 + 2x_3 = 15$

$2x_1 + x_2 + 4x_3 = 8$

11. Solve the system

$2x_1 + 4x_2 - 6x_3 = -8$

$x_1 + 3x_2 + x_3 = 10$

$2x_1 - 4x_2 - 2x_3 = -12$

using Gauss- Jordan method

ELECTIVE COURSE

MSCSC01DSE02PRINCIPLES OF PROGRAMMING AND NUMERICAL METHODS USING C

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

Numerical computational methods help to solve complex mathematical problems which cannot be solved easily by analytical mathematics by using simple arithmetic operations and which requires the use of an algorithm. The understanding of numerical computing helps to simulate various scientific models. The course will focus on different paradigms and approaches using C programming.

Course Objectives:

- To introduce basic concepts of Procedure Oriented Programming.
- Discuss about Errors and Approximations
- Discuss about the concept of numerical computational methods
- Discuss the numerical integration and differentiation:

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Acquainted with Numerical Methods.
CO2	Obtain the knowledge about Numerical Integration and Differentiation
CO3	Understanding the basic concepts of C programming.
CO4	Develop programming skill

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓		✓	✓	✓
CO2	✓	✓		✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓	✓

COURSE CONTENTS

Module 1: Introduction to Numerical Methods: Nature of numerical problems; computer based solutions. Errors and Approximations. Nonlinear equations – Bisection Method, Regular- Falsie Method, Newton- Raphson. System of Linear Equations- Gauss elimination, Gauss Jordan elimination, Triangulation method, Iterative method, Jacobi. Case study by writing algorithms.

Module 2: Numerical Integration and Differentiation: Concept of differentiation and Integration. Taylors series and Eulers methods-Simpson's Romberg, Gaussian, Runge-Kutta methods. Case study by writing algorithms.

Module 3: Algorithms and Flow charts: Definitions, Symbols, Program : structure, top-down design, source code, object code, executable file, file extensions. Importance of C; Basic structure of C, Programming style, executing a c program. Character set, C tokens, Keywords, identifiers, Constants, data types, declaration of variables, arithmetic operators , logical operators, Relational operators, Assignment operators, Increment and decrement operators, conditional operators, Bitwise operators. Precedence and order of evaluation. type conversion in expression. common programming errors, program testing and debugging, program efficiency. Managing Input output operation: reading a character, writing a character, formatted input output. Branching statements-if, if..else, nested if...else, else...if ladder, switch statement, go to statement. Looping statements- while, do...while, for loop. Break and continue statements.

Arrays: One dimensional arrays, two dimensional arrays, Initializing array elements, Multidimensional arrays. Strings: declaration and initializing, reading and writing. Arithmetic operations on character. String handling functions Functions: Library and user defined, defining a function, calling a function. Parameter passing techniques, Scope and life time of variables in function, recursive functions, arrays and functions.

Module 4: Structure and union: definition, giving values to members, initialization. Array of structures, array with in structure, structure with in structure, union. Pointers:

accessing the address of a variable, declaration and initializing pointers, accessing a variable through its pointers, pointer arithmetic, pointers and arrays (pointer to array and array of pointers) , pointers and character string , pointer and functions. Dynamic memory allocation: malloc(), calloc(), free(), realloc().

File Management: Text and binary files, Defining and opening a file, closing a file, input and output operations on file, error handling, random access file. Command line arguments.

Core Compulsory Readings

3. Discrete Mathematical Structures with Application to Computer Science-McGraw Hill
4. Introductory Methods of Numerical Analysis – January 2012 PHI, Sastry S.S

Core Suggested Readings

1. Balagurusamy, E., "Numerical Methods", Tata McGraw-Hill, New Delhi, 1999.
2. Computer Basics and c Programming, V. Rajaraman, PHI,2008
3. Let us C, Yeshvanth Kanethkar, 3rd Edn,BPB
4. C by Example, Noel Kalicharan, Cambridge Universitypress.

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Print

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%

Assignment	20%
Seminar…..	40%

Sample Questions to test Outcomes.

1. Write C programme to demonstrate Bisection method

2. Write the algorithm of False position method

3. Explain Gaussian elimination method with an example

4. Write the algorithm to implement Gauss-Jordan method

5. Write C program for computing the value of nP_r

6. Explain the concept of differentiation and Integration with examples.

7. Find the real root of the equation x^3-2x-5 = 0

8. Find the positive root, between 0 and 1, of the equation x = e^{-x} to a tolerance of 0.05%

9. Find a root, correct to three decimal places and laying between 0 and 0.5, of the equation

$4 e^{-x} \text{Sin } x - 1 = 0$

10. Obtain the solution of the following system using the Jacobi iteration method

$2x_1 + x_2 + x_3 = 5$

$3x_1 + 5x_2 + 2x_3 = 15$

$2x_1 + x_2 + 4x_3 = 8$

11. Solve the system

$2x_1 + 4x_2 - 6x_3 = -8$

$x_1 + 3x_2 + x_3 = 10$

$2x_1 - 4x_2 - 2x_3 = -12$

using Gauss- Jordan method

11. Write the algorithm for simulating Runge- Kutta method.

ELECTIVE COURSE

MSCSC01DSE03PRINCIPLES OF PROGRAMMING AND NUMERICAL METHODS USING C++

Credit			**Teaching Hours**			**Assessment Weightage (%)**		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

Numerical computational methods help to solve complex mathematical problems which cannot be solved easily by analytical mathematics by using simple arithmetic operations and which requires the use of an algorithm. The understanding of numerical computing helps to simulate various scientific models. The course will focus on different advanced paradigms and approaches using C++ programming.

Course Objectives:

- To introduce basic concepts of OOP
- Discuss about Errors and Approximations
- Discuss about the concept of numerical computational methods
- Discuss the numerical integration and differentiation

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Acquainted with Numerical Methods.
CO2	Obtain the knowledge about Numerical Integration and Differentiation
CO3	Understanding the basic concepts of C++ programming.
CO4	Develop skill in programming

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓		✓	✓	✓
CO2	✓	✓		✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓	✓

COURSE CONTENTS

Module 1: Introduction to Numerical Methods: Nature of numerical problems; computer based solutions. Errors and Approximations. Nonlinear equations – Bisection Method, Regular- Falsie Method, Newton- Raphson. System of Linear Equations- Gauss elimination, Gauss Jordan elimination, Triangulation method, Iterative method, Jacobi. Case study by writing algorithms.

Module 2: Numerical Integration and Differentiation: Concept of differentiation and Integration. Taylors series and Eulers methods-Simpson's Romberg, Gaussian, Runge-Kutta methods. Case study by writing algorithms.

Module 3: Principles of object oriented programming; OOP paradigm; Basic concepts of OOP; Benefits; applications. Introduction to C++, Structure of C++ program; Tokens, Keywords, identifiers and constants; Data types, symbolic constants; type compatibility; declaration and dynamic initialization of variables; reference variables.Operators, manipulators; type cast operators; Expressions, implicit conversions; operator overloading; operator precedence; Control structures.

Functions; function overloading; friend and virtual functions; Math library functions. Structures; Specifying a class; Defining member functions; making an outside function inline; nesting of member functions; private member functions; arrays within a class; memory allocation for objects; static data members; static member functions; arrays of objects; objects as function arguments; friendly functions; returning objects; const member functions; pointer to members; Local classes.

Module 4: Constructors and destructors; dynamic initialization of objects; copy constructor; Dynamic constructors; const objects; Destructors. Operator overloading – definition; overloading unary operators; overloading binary operators; overloading binary operators using friends; manipulation of strings using operators; rules for overloading operators. Type conversions.

Inheritance – defining derived classes; making a private member inheritance; Types of inheritance; virtual base classes; abstract classes; constructors in derived classes; Nesting of classes. Pointers; Pointers to objects; Pointers to derived classes; virtual functions; pure virtual functions.

C++ streams; stream classes; unformatted I/O operations; Formatted console I/O operations; Managing output with manipulators. Files – classes for file stream operations; Opening and closing a file; file modes; file pointers and their manipulations; Sequential input and output operation.

Core Compulsory Readings

5. Discrete Mathematical Structures with Application to Computer Science-McGraw Hill
6. Introductory Methods of Numerical Analysis – January 2012 PHI, Sastry S.S
7. Stroustrup, Bjarne (2000). Programming Principles and Practices using C++ (2nd Edition). Addison-Wesley. 2014

Core Suggested Readings

1. Balagurusamy, E., "Numerical Methods", Tata McGraw-Hill, New Delhi, 1999.

2. Balagurusamy, Object Oriented Prog With C++,5e, McGraw-Hill Education (India) Pvt Limited, 2011

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Print

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%

Assignment	20%
Seminar…..	40%

Sample Questions to test Outcomes.

1. Write C++ programme to demonstrate Bisection method

2. Write the algorithm of False position method

3. Explain Gaussian elimination method with an example

4. Write the algorithm to implement Gauss-Jordan method

5. Write C++ program for computing the value of nP_r

6. Explain the concept of differentiation and Integration with examples.

7. Find the real root of the equation $x^3-2x-5 = 0$

8. Find the positive root, between 0 and 1, of the equation $x = e^{-x}$ to a tolerance of 0.05%

9. Find a root, correct to three decimal places and laying between 0 and 0.5, of the equation $4 e^{-x} \text{Sin } x - 1 = 0$

10. Obtain the solution of the following system using the Jacobi iteration method

$2x_1 + x_2 + x_3 = 5$

$3x_1 + 5x_2 + 2x_3 = 15$

$2x_1 + x_2 + 4x_3 = 8$

11. Solve the system using Gauss- Jordan method

$2x_1 + 4x_2 - 6x_3 = -8$

$x_1 + 3x_2 + x_3 = 10$

$2x_1 - 4x_2 - 2x_3 = -12$

12. Write the algorithm for simulating Runge- Kutta method.