SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

(Deemed to be University U/S 3 of UGC Act 1956) (Accredited with "A" Grade by NAAC)
Enathur, Kanchipuram – 631561

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CURRICULUM AND SYLLABUS FOR FULL TIME

B.E. (Computer Science and Engineering)

(Applicable for students admitted from 2025-2026 onwards)

REGULATION AY 2025-2026

B.E. COMPUTER SCIENCE AND ENGINEERING

These regulations are applicable to the students admitted from the AY 2025-26 Onwards.

CHOICE BASED CREDIT SYSTEM FOR B.E. (CSE) FULL-TIME PROGRAMME CREDITS

Theory courses: Courses with 4/3 credits will be assigned 4/3 Lecture and 0/1 Tutorial hours per week.

Practical courses: Courses with 2 credits will be assigned 4 hours of lab/practical work per week

Each semester curriculum shall normally have a blend of theory and practical courses. In the first year the total number of credits will be 38. From semester III to VII, the average credits per semester will be 21 and for semester VIII, the credits will be 16. For the award of the degree, a student has to earn a minimum of 163 credits.

DURATION OF THE PROGRAMME

A student is normally expected to complete B.E (CSE) programme in four years and in any case, not more than seven years from the time of admission.

REGISTRATION FOR COURSES

A newly admitted student will automatically be registered for all the courses prescribed for the first year, without any option.

All other students shall submit a completed registration form indicating the list of courses intended to be credited during the next semester. This registration will be done a week before the last working day of the current semester. Late registration, with the approval of the Dean on the recommendation of the Head of the Department, along with a late fee will be done, up to the last working day.

Registration for the project work shall be done only for the final semester.

ASSESSMENT

The break-up of assessment and examination marks for theory subjects is as follows.

First Assessment Test	:	15 Marks
Second Assessment Test	:	15 Marks
Assignment & Attendance (seminars, group discussion)	:	10 Marks
Total (Internal Marks)	:	40 Marks
End semester Examination (External Marks)	:	60 Marks
Total (Internal + External)	:	100 Marks

The break-up of the assessment and examination marks for practical is as follows.

Observations	:	15 Marks
Model Test	:	15 Marks
Record book & Attendance	:	10 Marks

Total (Internal Marks)	:	40 Marks
End semester Examination (External Marks)	:	60 Marks
Total (Internal + External)	:	100 Marks

The project work will be assessed for 40 marks by a committee consisting of the Guide and the Head of the Department. The Head of the Department shall be the Chairman. 60 marks are allotted for the project viva voce examination at the end of the semester.

WITHDRAWAL FROM A COURSE

A student can withdraw from the course at any time before a date fixed by the Head of the Department prior to the second assessment, with the approval of the Dean on the recommendation of the Head of the Department.

TEMPORARY BREAK OF STUDY

A student can take a one-time temporary break of study covering the current year/semester and/or the next semester with the approval of the Dean on the recommendation of the Head of the Department, not later than seven days after the completion of the mid-semester test. However, the student must complete the entire program within the maximum period of seven years.

SUBSTITUTE ASSESMENT

A student, who has missed, for genuine reasons accepted by the Head of the Department, one or more of the assessments of a course other than the end semester examination, may take a substitute assessment for any one of the missed assessments. The substitute assessment must be completed before the commencement of the end-semester examination.

A student who wishes to have a substitute assessment for a missed assessment must apply to the concerned faculty member within a week from the date of the missed assessment.

ATTENDANCE REQUIREMENTS

To be eligible to appear for the examination in a particular course, a student must put in a minimum of 80% of attendance in the course. However, if the attendance is 70% or above but less than 80% in any course, the authorities can permit the student to appear for the examination in the course on payment of the prescribed condonation fee.

A student who withdraws from or does not meet the minimum attendance requirement in the course must re-register for and repeat the course.

PASSING AND DECLARATION OF EXAMINATION RESULTS

All assessments of all the courses on the absolute mark basis will be considered and passed by the results passing board in accordance with the rules of the University. Thereafter, the Controller of Examinations shall convert the marks for each course to the corresponding letter grade as follows, compute the grade point average & cumulative grade point average and prepare the grade cards.

90 to 100 marks	-	Grade 'S'
80 to 89 marks	-	Grade 'A'
70 to 79 marks	-	Grade 'B'
60 to 69 marks	-	Grade 'C'
55 to 59 marks	-	Grade 'D'
50 to 54 marks	-	Grade 'E'
less than 50 marks	-	Grade 'F'
Insufficient attendance	-	Grade 'I'
Withdrawn from the course	-	Grade 'W'

A student who obtains less than 50 marks out of 100 in the subject or less than 24 out of 60 in External exam or is absent for the examination will be awarded Grade 'F'.

A student who earns a grade of S, A, B, C, D or E for a course is declared to have successfully completed that course and earned the credits for that course. Such a course cannot be repeated by the student.

A student who obtains letter grade F in a course has to reappear for the examination in that course.

The following grade points are associated with each letter grade for calculating the grade point average.

S – 10; A-9; B-8; C-7; D-6; E-5; F-0

A student can apply for revaluation of one or more of his /her examination answer papers within a week from the date of issue of Grade sheet to the student on payment of the prescribed fee per paper. The application must be made to the Controller of Examinations with the recommendation of the Head of the Department.

After results are declared, Grade cards will be issued to the students. The Grade card will contain the list of courses registered during the year/semester, the grades scored and the grade point average (GPA) for the year/semester.

GPA is the sum of the products of the number of credits of a course with the grade point scored in that course, taken over all the courses for the Year/Semester , divided by the sum of the number of credits for all courses taken in that year/semester. CGPA is similarly calculated considering all the courses taken from the time of admission.

After successful completion of the program, the Degree will be awarded with the following classification based on CGPA:

For First Class with Distinction, the student must earn a minimum of 163 credits within four years from the time of admission, pass all the courses in the first attempt and obtain a CGPA of 8.25 or above.

For First Class, the student must earn a minimum of 163 credits within four years from the time of admission and obtain a CGPA of 6.5 or above.

For Second Class, the student must earn a minimum of 163 credits within seven years from the time of admission.

ELECTIVES

Apart from the various Core courses offered in the curriculum of the branch of specialization, a student can choose electives from a list of electives offered by the Department and from other Departments with the approval of the Head of the Department and the Head of the Department offering the course.

EXAMINATION PATTERN FOR MANDATORY COURSES

There will not be any External examination for Mandatory Courses. Performance of students will be assessed through tests and assignments conducted by the Department. The internal assessment pattern is as follows.

First test	: 30 Marks
Second test	: 30 Marks
Assignment (G.D + Seminar + Attendance + Class test)	: 40 Marks
	----------
Total Marks	: 100 Marks
Passing Minimum marks	: 50%

EXAMINATION PATTERN FOR INDUSTRIAL AND TRAINING PRACTICE

There will be external examination for Industrial and Training Practice. Performance of students will be assessed through offline and online internship in various companies along with presentation.

The internal and external assessment pattern is as follows.

Internal Presentation and Certificates	: 40 Marks
External Presentation	: 60 Marks
Total Marks	: 100Marks
Passing Minimum marks	: 50%

Marks are allotted for continuous reviews and internal presentation (40) and external presentation (60). A Candidate shall be declared to have passed the examination, if he/she has secured a minimum mark of 50%.

MAPPING OF PEOS, POS and PSOs

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. Provide engineering insight to problem solving to succeed in Technical Profession through precise education and to prepare students to excel in postgraduate programs.
2. To provide students with fundamental knowledge and ability to expertise in Computer Science and Engineering.
3. III. Prepare students with good scientific and engineering breadth so as to analyze, design and create products, solutions to problems in the area of Computer Science and Engineering.
4. IV. To inculcate in students professional, effective communication skills, team work, multidisciplinary approach and an ability to relate engineering issues to broader social context.
5. Prepare students to be aware of excellence, leadership, written ethical codes and guidelines and lifelong learning needed for successful professional career by providing them with an excellent academic environment.

PROGRAM OUTCOME(S) (POs) for B.E (CSE)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify, formulate, review research literature, and analyzes complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
3. **Design/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (Pso) for B.E.(CSE)

- To apply fundamental knowledge of computing and techniques to develop more efficient and effective in software, hardware mechanisms.
- To analyze, design, implement, and evaluate a computational system to meet desired needs within dynamic realistic constraints.
- To apply innovative ideas into reality, enhancing research capability, ethical and entrepreneurial practice.

DEFINITION OF CREDIT

Sl.No	No of hours per week	No of credits
1.	1 Hr. Lecture (L) per week	1
2.	1 Hr. Tutorial (T) per week	1
3.	1 Hr. Practical (P) per week	0.5

CREDIT DISTRIBUTION

Sl.No	Category	Credit
1.	Humanities and Social Science Including Management Courses (HSMC)	12
2.	Basic Science Courses(BSC)	24
3.	Engineering Science Courses(ESC)	25
4.	Professional Core Courses(PCC)	68
5.	Professional Elective Course(PEC)	12
6.	Open Elective Courses(OEC)	6
7.	Mandatory Courses(MC)	0
8.	Industrial Training & Practice(INT.)	1
9.	Project Work(PROJ)	15
Total Credits		**163**

COURSE CODE AND DEFINITION

Course Code	Definitions
L	Lecture
T	Tutorial
P	Practical
IA	Internal Assessment
EA	External Assessment
TM	Total Marks
HSMC	Humanities and Social Science Including Management Courses
BSC	Basic Science Courses
ESC	Engineering Science Courses
PCC	Professional Core Courses
PEC	Professional Elective Course
OEC	Open Elective Courses
MC	Mandatory Courses
INTE.	Industrial Training & Practice
PROJ.	Project Work

SUMMARY OF CREDITS

Category/ Semester	HSMC	BSC	ESC	PCC	PEC	OEC	MC	Inte.	Proj.	Total
I		8	12	0						**20**
II	6	8	4	0						**18**
III	1	4	4	14						**23**
IV	3	4	5	10			**NC**			**22**
V	1			21						**22**
VI	1			13	6		**NC**	1	2	**23**
VII				10	3	3	**NC**		3	**19**
VIII				0	3	3			10	**16**
Total	**12**	**24**	**25**	**68**	**12**	**6**	**NC**	**1**	**15**	**163**

NC – Non Credit

CURRICULUM OF B.E.(CSE)

SEMESTER – III									
Sl. No	Course Type	Course Title	Hours per Week			C	IA	EA	TM
			L	T	P				
1		Probability and Statistics	4	0	0	4	40	60	100
2		Digital Logic Design	3	0	2	4	40	60	100
3		Computer Organization and Architecture	3	0	2	4	40	60	100
4		Data Structures and Algorithms	3	0	4	5	40	60	100
5		Fundamentals of Machine Learning	3	0	4	5	40	60	100
6		Humanities (Soft Skills-II)	1	0	0	1	100	-	100
Total						**23**	**Credit: 63/163**		

SEMESTER – IV									
Sl. No	Course Type	Course Title	Hours per Week			C	IA	EA	TM
			L	T	P				
1		Discrete Mathematics	4	0	0	4	40	60	100
2		Fundamentals of Embedded Systems and IoT	3	0	4	5	40	60	100
3		Design and Analysis of Algorithm	3	0	4	5	40	60	100
4		Object Oriented Programming	3	0	4	5	40	60	100
5		Management – I (Organizational Behavior)	3	0	0	3	40	60	100
6		Mandatory Course (Indian Knowledge System)	1	0	0	NC	-	-	-
Total						**22**	**Credit: 85/163**		

SEMESTER – V									
Sl. No	Course Type	Course Title	Hours per Week			C	IA	EA	TM
			L	T	P				
1		Theory of Computation	3	0	0	3	40	60	100
2		Operating System	3	0	4	5	40	60	100
3		Database Systems	3	0	4	5	40	60	100
4		Computer Networks	3	0	4	5	40	60	100
5		Introduction to Cyber Security	3	0	0	3	40	60	100
6		Entrepreneurship & Start-ups	2	0	0	1	40	60	100
Total						**22**	**Credit: 107/163**		

SEMESTER – VI									
Sl. No	Course Type	Course Title	Hours per Week			C	IA	EA	TM
			L	T	P				
1		Compiler Design	3	0	4	5	40	60	100
2		Data Mining	3	0	4	5	40	60	100
3		Software Engineering	3	0	0	3	40	60	100
4		Professional Elective-I	3	0	0	3	40	60	100
5		Professional Elective-II	2	0	2	3	40	60	100
6		Creative and Innovative Project	0	0	4	2	40	60	100
7		Industrial Training and Practices	0	0	2	1	40	60	100
8		Humanities (Soft Skills-II)	1	0	0	1	100	-	100
9		Mandatory Course (Environmental Science)	0	0	0	NC	-	-	-
Total						**23**	**Credit: 130/163**		

SEMESTER – VII									
Sl. No	Course Type	Course Title	Hours per Week			C	IA	EA	TM
			L	T	P				
1		Web Technologies	3	0	4	5	40	60	100
2		Professional Elective-III	3	0	0	3	40	60	100
3		Open Elective-I	3	0	0	3	40	60	100
4		Mobile Application Development	3	0	4	5	40	60	100
5		Project Work Phase-I	0	0	6	3	40	60	100
6		Mandatory Course (Constitution of India)	0	0	0	NC	-	-	-
Total						**19**	**Credit: 149/163**		

SEMESTER – VIII									
Sl. No	Course Type	Course Title	Hours per Week			C	IA	EA	TM
			L	T	P				
1		Professional Elective-IV	3	0	0	3	40	60	100
2		Open Elective-II	3	0	0	3	40	60	100
3		Project Work Phase -II	0	0	20	10	40	60	100
Total						**16**	**Credit: 165/163**		

Humanities and Social Science Including Management Courses (HSMC)						
Sl.No	**Category**	**Course Title**	**L**	**T**	**P**	**C**
1.	HSMC	English	2	0	0	2
2.	HSMC	Universal Human Values	3	0	0	3
3.	HSMC	Humanities (Soft Skills-I)	1	0	0	1
4.	HSMC	Humanities (Soft Skills-II)	1	0	0	1
5.	HSMC	Humanities (Soft Skills-III)	1	0	0	1
6.	HSMC	Management – I (Organizational Behavior)	3	0	0	3
7.	HSMC	Entrepreneurship & Start-ups	2	0	0	1
Total						**12**

BASIC SCIENCE COURSES(BSC)						
Sl.No	**Category**	**Course Title**	**L**	**T**	**P**	**C**
1.	BSC	Engineering Mathematics - I	4	0	0	4
2.	BSC	Engineering Physics	3	0	2	4
3.	BSC	Engineering Mathematics – II	4	0	0	4
4.	BSC	Engineering Chemistry	3	0	2	4
5.	BSC	Probability and Statistics	4	0	0	4
6.	BSC	Discrete Mathematics	4	0	0	4
Total						**24**

Engineering Science Courses (ESC)						
Sl. No	**Category**	**Course Title**	**L**	**T**	**P**	**C**
1.	ESC	Basic Electrical Engineering	3	0	2	4
2.	ESC	Essentials of Computing	3	0	2	5
3.	ESC	Engineering Graphics & Design	2	0	2	3
4.	ESC	Design Thinking & IDEA Lab	2	0	2	3
5.	ESC	Programming for Problem Solving	3	0	4	5
6.	ESC	Digital Logic Design	3	0	2	4
7.	ESC	Fundamentals of Embedded Systems and IoT	3	0	4	5
Total						**29**

PROFESSIONAL CORE COURSES(PCC)						
Sl.No	**Category**	**Course Title**	**L**	**T**	**P**	**C**
1.		Computer Organization and Architecture	3	0	2	4
2.		Data Structures and Algorithms	3	0	4	5
3.		Fundamentals of Machine Learning	3	0	4	5
4.		Design and Analysis of Algorithm	3	0	4	5
5.		Object Oriented Programming	3	0	4	5
6.		Theory of Computation	3	0	0	3
7.		Operating System	3	0	4	5
8.		Database Systems	3	0	4	5
9.		Computer Networks	3	0	4	5
10.		Introduction to Cyber Security	3	0	0	3
11.		Compiler Design	3	0	4	5
12.		Data Mining	3	0	4	5
13.		Software Engineering	3	0	0	3
14.		Web Technologies	3	0	4	5
15.		Mobile Application Development	3	0	4	5
Total						**68**

PROFESSIONAL ELECTIVE COURSE(PEC)						
Sl.No	**Category**	**Course Title**	**L**	**T**	**P**	**C**
1.	PEC	Professional Elective-I	3	0	0	3
2.	PEC	Professional Elective-II	2	0	2	3
3.	PEC	Professional Elective –III	3	0	0	3
4.	PEC	Professional Elective –IV	3	0	0	3
Total						**12**

OPEN ELECTIVE COURSES(OEC)

Sl.No	Category	Course Title	L	T	P	C
1.	OEC	Open Elective – I	3	0	0	3
2.	OEC	Open Elective –II	3	0	0	3
Total						**6**

MANDATORY COURSES(MC)						
Sl. No	**Category**	**Course Title**	**L**	**T**	**P**	**C**
1.		Indian Knowledge System	1	0	0	**NC**
2.		Environmental Science	1	0	0	**NC**
3.		Constitution of India	1	0	0	**NC**

PROFESSIONAL ELECTIVES	
SEMESTER–VI	
Subject Code	**Subject Name**
Professional Elective–I	
	Software Architecture
	DevOps Engineering
	Software Project Management
	Software Testing
	Advanced Programming
Professional Elective–II	
	Deep Learning
	Natural Language Processing
	Business Intelligence & Data Warehousing
	Soft Computing
	Computer Vision
SEMESTER– VII	
Professional Elective–III	
	Cloud Computing
	Distributed Systems
	Parallel Computing
	Big data Analytics
	Quantum Computing
SEMESTER–VIII	
Professional Elective–IV	
	Augmented & Virtual Reality
	Data Visualization
	Speech and Pattern Recognition (Sensory Technologies)
	UX/UI Design
	Software Design Thinking

OPEN ELECTIVES	
SEMESTER– VII	
Subject Code	**Subject Name**
Open Elective – I	
	Engineering Economics
	Principles of Management
	Computational Biology
	Financial Computing
	Smart Cities and Sustainability
	Disaster Management
SEMESTER– VIII	
Open Elective – II	
	Business Analytics
	Professional Ethics
	Health Informatics
	Fintech
	Assistive Technology
	Smart Agriculture
	Research methodology and IPR

III – SEMESTER

Course Code		L	T	P	C
Course Title	PROBABILITY AND STATISTICS	4	0	0	4

PRE-REQUISITES

OBJECTIVES

•	To introduce fundamental concepts of probability theory, including conditional probability, Bayes' theorem, and random variables.
•	To familiarize students with discrete and continuous probability distributions and their applications.
•	To develop an understanding of statistical measures such as central tendency, dispersion, correlation, and regression.
•	To equip students with techniques for curve fitting and conducting large sample hypothesis testing.
•	To enable students to perform small sample hypothesis tests using t, F, and chi-square distributions.

COURSE OUTCOMES

At the end of course, the students will be able to

1.	Apply basic concepts of probability, including Bayes' theorem and moment generating functions, to analyze random events.
2.	Use standard discrete and continuous probability distributions to model and solve real-world problems.
3.	Compute and interpret statistical measures such as mean, variance, correlation, regression, skewness, and kurtosis.
4.	Perform curve fitting using least squares and conduct large-sample hypothesis testing.
5.	Apply small-sample tests such as t-test, F-test, and chi-square test for inference and decision-making.

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01															
CO 02															
CO 03															
CO 04															
CO 05															

LEGEND: 1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	PROBABILITY	12

Introduction to Probability, Probability spaces, conditional probability, Bayes' Theorem, Discrete and Continuous one dimensional random variables - Expectations, Moments, Variance of a sum, Moment generating function, Tchebyshev's Inequality.

UNIT – II	PROBABILITY DISTRIBUTIONS	12

Discrete Distributions – Binomial, Poisson and Negative Binomial distributions, Continuous Distributions - Normal, Exponential and Gamma distributions.		
UNIT – III	**STATISTICS**	**12**
Measures of Central tendency, Measures of dispersion, co efficient of variation, Moments, Skewness and Kurtosis, Correlation, Rank Correlation and Regression (Bivariate)		
UNIT – IV	**TESTING OF HYPOTHESIS-I**	**12**
Curve fitting by the method of least squares- fitting of straight lines, second degree parabolas and more general curves. Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations.		
UNIT – V	**TESTING OF HYPOTHESIS-II**	**12**
Test for single mean, difference of means and correlation coefficients, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes.		
TEXT BOOKS		
1.	T. Veerarajan, Probability, Statistics and Random Processes, Third edition, Tata McGraw- Hill, New Delhi, 2010.	
2.	S.P. Gupta, Statistical Methods, 31st edition, Sultan chand and sons, New Delhi, 2002.	
REFERENCES		
1.	Loeve, M. (2012). Probability Theory I. United States: Springer New York.	
PREPARED BY		
Dept. of Mathematics		

Course Code		L	T	P	C
Course Title	DIGITAL LOGIC DESIGN	3	0	2	4

PRE-REQUISITES

Basic Electrical and Electronics Engineering

OBJECTIVES

- To understand and apply various number systems for solving simple conversion and arithmetic problems.
- To simplify mathematical expressions using Boolean algebra and Boolean functions for basic digital logic problems.
- To understand combinational circuits using standard digital design techniques.
- To design synchronous sequential circuits and registers.
- To understand various memory devices and design digital circuits using HDL programming.

COURSE OUTCOMES

At the end of course, the students will be able to

1.	Apply various number systems and perform arithmetic operations for digital computation.
2.	Simplify Boolean expressions using Boolean algebra and Karnaugh maps for digital logic applications.
3.	Design and implement combinational logic circuits for digital applications.
4.	Design synchronous sequential circuits and registers for digital system operations.
5.	Develop HDL programs to model digital circuits and examine various memory devices used in digital systems.

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	L	M	M	H				M					3	2	1
CO 02	L	M	M	H			L	M					3	2	1
CO 03	L	M	L	H				L	L				3	2	2
CO 04	L	H	M	M		M		L	M	M			3	2	2
CO 05	L	M	M	M		M		M	M				3	2	2

LEGEND: 1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	NUMBER SYSTEMS	9

Number systems: Decimal, Binary, Octal, Hexadecimal - Codes: ASCII code, Gray Code, BCD code - Complements - Error Detection and Error Correction.

UNIT – II	BOOLEAN ALGEBRA	9

Truth Table - theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms, K-map (upto 4 Variables), Don't care conditions - Minimization and Quine-McCluskey method of minimization.

UNIT – III	COMBINATIONAL CIRCUITS	9
Digital logic gates - Design procedure - Universal Gates – Adder – Subtractor - Encoder – Decoder – Multiplexer – Demultiplexer.		
UNIT – IV	**SYNCHRONOUS SEQUENTIAL CIRCUITS AND REGISTERS**	**9**
SEQUENTIAL CIRCUITS: Flip flop – SR Flip flop – D Flip flop - JK Flip flop - Master Slave Flip Flop. REGISTERS AND COUNTERS: Registers– Shift Register – Ripple Counter – Synchronous Counter.		
UNIT – V	**LOGIC FAMILIES AND PROGRAMMABLE LOGIC DEVICES**	**9**
HDL Programming: Introduction to HDL Programming, HDL for Combinational Circuits, HDL for sequential logic circuits. Programmable Logic Devices: Programmable Logic Array (PLA)-Programmable Array Logic (PAL) - PROM.		

LIST OF EXPERIMENTS

UNIT – I	**NUMBER SYSTEMS**
1.	Verification of truth tables of the Logic gates.
2.	Design and implement code conversion (Binary to Gray and Gray to Binary code converters, BCD to Excess-3 and Excess-3 to BCD)
UNIT – II	**BOOLEAN ALGEBRA**
3.	Implementation the Boolean function F (A, B, C) = A'B'C + ABC' + AB'C using logic gates
4.	Verify Demorgan's Theorem for 2 variables using logic gates.
UNIT – III	**COMBINATIONAL CIRCUITS**
5.	Design and Implementation of Full Adder and Full Subtractor Circuits using logic gates.
6.	Design and Implementation of a function using Multiplexer and Demultiplexer.
UNIT – IV	**SYNCHRONOUS SEQUENTIAL CIRCUITS AND REGISTERS**
7.	Design and verify the operation of D Flip-Flop and JK Flip-Flop
8.	Design and implement SISO, PIPO/PISO, SIPO shift registers and verify their outputs.
UNIT – V	**LOGIC FAMILIES AND PROGRAMMABLE LOGIC DEVICES**
9.	Design of logic gates using HDL and verify their outputs.
10.	Design Adder/ Subtractor using HDL programming.

Digital Trainer Kit, Xilinx Software , HDL

TEXT BOOKS

1.	M. Morris Mano, "Digital Design", 6th Edition, Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2018.
2.	W.H. Gothmann, "Digital Electronics - An Introduction, Theory and Practice",Prentice Hall of India, 2nd Edition 2000
3.	Frank Vahid "VHDL for Digital Design-With RTL design, VHDL & Verilog"- John Wiley & Sons, 2010
4.	R. P. Jain and Kishor Sarawadekar, "Modern Digital Electronics", McGraw Hill Education, 5th Edition, 2022.

REFERENCES	
1.	A. Anand Kumar, "Fundamentals of Digital Circuits", PHI Learning Private Limited, 4th Edition, 2021.
2.	A. P. Godse and D. A. Godse, "Digital IC Applications", Technical Publications, 1st Edition, 2007.
3.	D. J. Comer, "Digital Logic and State Machine Design", Oxford University Press Inc., 3rd Edition, 2012.
4.	Thomas L. Floyd and R. P. Jain, "Digital Fundamentals", Pearson Education, 11th Edition, 2017.
5.	John F. Wakerly, "Digital Design: Principles and Practices", Pearson Education / PHI, 4th Edition, 2008.
6.	Charles H. Roth Jr. and Larry L. Kinney, "Fundamentals of Logic Design", Cengage / Thomson Learning, 6th Edition, 2019.
7.	R. Anand, "Digital Electronics", Khanna Publishing House, 2018.
LEARNING OUTCOMES	
By the end of this course, learners will be able to:	
1.	Understanding of number systems and solve related arithmetic and conversion problems.
2.	Analyze and simplify Boolean expressions for digital logic design applications.
3.	Design combinational circuits using standard digital design methodologies.
4.	Design synchronous sequential circuits for digital system operations.
5.	Develop HDL programs for digital circuit implementation and examine memory devices in digital systems.
PREPARED BY	
Department of ECE & Coordinated by Dr.T.Lakshmibai	

Course Code		L	T	P	C
Course Title	COMPUTER ORGANIZATION & ARCHITECTURE	3	0	2	4

PRE-REQUISITES

Programming Fundamentals and Data Representation

OBJECTIVES

- Understand the fundamentals of computer organization and architecture
- Analyze the design and operation of CPU and instruction execution
- Learn data representation and arithmetic operations in computers
- Examine input/output organization and data transfer mechanisms
- To describe memory hierarchy and concept of virtual memory.

COURSE OUTCOMES

At the end of course, the students will be able to

1.	Explain the basic structure and operation of a computer system, including instruction execution and control mechanisms.
2.	Analyze the organization and functioning of the CPU, including registers, instruction formats, and addressing modes.
3.	Apply arithmetic algorithms and data representation techniques used in computer systems.
4.	Evaluate input/output organization and data transfer methods, including interrupt and DMA techniques.
5.	Understand the basics of memory systems and its type's.

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO0 5	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	3	2	-	-	-	-	-	-	-	-	-	1	3	-	-
CO 02	3	3	1	-	-	-	-	-	-	-	-	1	3	1	-
CO 03	3	2	2	-	-	-	-	-	-	-	-	1	3	2	-
CO 04	3	2	1	-	1	-	-	-	-	-	-	1	2	2	1
CO 05	3	3	2	-	1	-	-	-	-	-	-	2	3	2	2

LEGEND:1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	BASIC COMPUTER ORGANIZATION AND DESIGN	9

Instruction Code, Operation Code, Stored Program Concept, Registers and memory of Basic Computer, Common Bus System for Basic Computer, Instruction Format, Control Unit of Basic Computer. Instruction Cycle of Basic computer, Determining Type of Instruction, Memory Reference Instructions, Input- Output Instructions.

UNIT – II	CENTRAL PROCESSING UNIT	9

General register organization, Stack organization, Instruction format, Addressing Modes(Implied Mode, Immediate Mode, Register Mode, Register Indirect Mode, Auto

increment or Auto decrement Mode, Direct Address Mode, Indirect Address Mode), Data transfer and manipulation, program control, Reduced Instruction Set Computer (RISC) - CISC Characteristics, RISC Characteristics		
UNIT – III	**COMPUTER ARITHMETICS**	**9**
Addition and subtraction, Multiplication and division algorithm, Floating point arithmetic Operations, decimal arithmetic unit.		
UNIT – IV	**INPUT OUTPUT ORGANIZATION**	**9**
Input-Output Interface: I/O Bus and Interface Modules, I/O vs. Memory Bus, Isolated vs. Memory-Mapped I/O, Asynchronous Data Transfer: Strobe, Handshaking, Modes of Transfer: Programmed I/O, Interrupt-Initiated I/O, Direct memory Access, Priority Interrupt: Polling, Daisy-Chaining, Parallel Priority Interrupt, Direct Memory Access, Input- Output Processor, DMA vs. IOP		
UNIT – V	**MEMORY ORGANIZATION**	**9**
Memory Hierarchy, Main Memory, RAM and ROM Chips, Memory address Map, Memory Connection to CPU, Auxiliary Memory (Magnetic Disk, Magnetic Tape), Associative Memory: Hardware Organization, Match Logic, Read Operation, Write Operation, Cache Memory: Locality of Reference, Hit & Miss Ratio, Mapping, Write Policies, virtual memory.		
TEXT BOOKS		
1	M. Morris Mano, "Computer System Architecture", Prentice-Hall of India, Pvt. Ltd., Third edition(Revised), 2017.	
2	Carl Hamacher, ZvonkoVranesic, SafwatZaky, NaraigManjikian, "Computer Organization and Embedded Systems", McGraw-Hill Publishing, 2011	
REFERENCES		
1	William Stallings, "Computer Organization and Architecture", Prentice-Hall of India, Pvt. Ltd., Eleventh edition, 2019.	
2	Vincent P. Heuring and Harry F. Jordan, "Computer System Design and Architecture", Prentice-Hall of India, Pvt. Ltd., Second edition, 2003.	
3	Andrew S. Tanenbaum and Todd Austin, "Structured Computer Organization", Pearson Education, 6th Edition, 2013 / Reprint 2016.	
4	David A. Patterson and John L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface", Morgan Kaufmann / Elsevier, 6th Edition, 2020.	
LIST OF EXPERIMENTS		
UNIT – I	**BASIC COMPUTER ORGANIZATION AND DESIGN**	
1.	Demonstration of Personal Computer Components, Dismantling and Assembly.	
2.	To design and simulate Common Bus System for Basic Computer.	
UNIT – II	**CENTRAL PROCESSING UNIT**	
3.	To simulate AND, OR, NOT, NAND, NOR Gates.	
4.	To simulate Encoder/Decoder.	
UNIT – III	**COMPUTER ARITHMETICS**	
5.	To design and simulate ALU unit for Booth Multiplication Algorithm.	
6.	To design and simulate ALU unit for Division algorithm.	
UNIT – IV	**INPUT OUTPUT ORGANIZATION**	

7.	To design and simulate Handshaking Data Transfer.
8.	To design and simulate a Direct Memory Access (DMA) controller.
UNIT – V	**MEMORY ORGANIZATION**
9.	To design and simulate Associative Memory with match logic.
10.	To design and simulate Memory for Write and Read instruction.
REFERENCES	
1.	"Verilog HDL: A guide to Digital Design and Synthesis" - Samir Palnitkar , SunSoft Press 1996.
WEB REFERENCES	
1.	http://download.xilinx.com/direct/ise9_tutorials/ise9tut.pdf
2.	http://web.stanford.edu/class/ee183/handouts_win2003/Modelsim_short_tutorial.pdf
3.	http://bertrand.granado.free.fr/Sysprog/SysProg/Cours_files/modelsim_tut.pdf
Programming Environment: Model Sim – Altera 10.3D(QUARTUS –II 15.0)	
LEARNING OUTCOMES	
1.	Understand the basic organization of a computer system, instruction cycle, and control unit operations.
2.	Analyze CPU architecture, including registers, instruction formats, and addressing modes.
3.	Apply arithmetic operations and algorithms used for data processing in computer systems.
4.	Explain input-output organization and data transfer techniques such as interrupts and DMA.
5.	Understand memory organization, hierarchy, cache, and virtual memory concepts.
PREPARED BY	
Dr. T. Sundar, AP/CSE & Dr. N C A. Boovarahan, AP/CSE	

Course Code		L	T	P	C
Course Title	DATA STRUCTURES AND ALGORITHMS	3	0	4	5

PRE-REQUISITES

Basic knowledge in C Programming & problem-solving skills

OBJECTIVES

•	To understand the principles of algorithmic efficiency and the practical trade-offs between time and space.
•	To understand fundamental concepts of searching and basic sorting techniques to build a foundation for problem-solving.
•	To explain and implement linear and non-linear data structures including stacks, queues, lists, trees, and graphs.
•	To develop the ability to select and design appropriate data structures for common engineering scenarios.

COURSE OUTCOMES

At the end of course, the students will be able to

1.	**Describe** algorithm efficiency qualitatively and understand the impact of data size on performance
2.	**Implement** searching algorithms and evaluate their practical behavior in different scenarios.
3.	**Design and implement** linear data structures and demonstrate their core operations.
4.	**Understand** and **implement** tree data structures (Binary Trees, BST) and explain the necessity of balanced trees (AVL, B-Tree).
5.	**Implement** graph traversals, basic sorting techniques, and hashing methods for efficient data processing.

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	3	3	1	1	1	–	–	–	–	–	–	2	3	2	1
CO 02	3	3	2	2	2	–	–	–	–	–	–	1	3	3	1
CO 03	3	3	3	2	2	–	–	–	2	1	–	1	3	3	2
CO 04	3	3	3	2	2	–	–	–	1	–	–	2	3	3	2
CO 05	3	3	2	2	2	–	–	–	1	–	–	2	3	3	2

LEGEND:1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	INTRODUCTION	9

Basic Terminologies & Data Structures - Elementary Data Organizations, Primitive and non-primitive. Array data structure - Single and multi-dimensional arrays; Memory Representation. Data Structure Operations - Insertion, deletion, traversal. Basics of Algorithm - Definition, Characteristics, Algorithm Efficiency (No notations). Time vs. Space trade-offs.

UNIT – II	STACKS AND QUEUES	9

Abstract Data Types (ADT) - Introduction to ADT & its importance. ADT Stack - PUSH, POP, PEEK Algorithms. Stack Applications - List of applications, Infix to Postfix and Parentheses Balancing algorithms using array. ADT Queue - Algorithms of ENQUEUE, DEQUEUE in Simple Queue, Circular Queue. Applications of Queues. Priority Queue - FindMin, Insert, DeleteMin, Implementation using unordered array.		
UNIT – III	**LINKED LISTS**	**9**
Linked List Basics - Linked List Representation in memory, Comparison with Arrays. Linked List Algorithms - Insertion (Beginning, Position, End), Deletion (Beginning, Position, End) and Traversing for Singly linked list & Doubly linked list. Implementation of Stack using Linked List.		
UNIT – IV	**TREES**	**9**
Tree Basics - Introduction & Terminology. Binary Tree - Introduction and Insertion, Deletion, Level-Order, In-Order, Pre-Order and Post-Order Algorithms. Binary Search Tree - Introduction and Insertion, Deletion, Level-Order, In-Order, Pre-Order and Post-Order Algorithms. Applications of Binary Tree & BST. AVL Tree - Introduction and necessity of Rotations, AVL Tree applications. Introduction to B-Tree - What is B-Tree and its applications.		
UNIT – V	**GRAPHS, SORTING AND HASHING**	**9**
Graph Basics - Introduction & Terminology. Traversal Algorithms - BFS & DFS. Sorting Algorithms - Selection, Bubble, Quick & Merge. Introduction to Heap and Heap Sort using Arrays. Hashing - Hash Functions-Separate Chaining, Open Addressing, Collision		
LIST OF EXPERIMENTS		
UNIT – I	**INTRODUCTION**	
1.	Implementation of Basics Array Operations (Insertion, Deletion, Traversal)	
2.	Building a Dynamic Employee Record System using Structure Pointers	
UNIT – II	**STACKS AND QUEUES**	
3.	Implementation of stack and expression conversion - Infix To Postfix	
4.	Postfix Expression evaluation using Stack	
5.	Implementation of Simple Queue using Array	
6.	Implementation of Priority Queue using unordered Array	
UNIT – III	**LINKED LISTS**	
7.	Implementation of Singly Linked List (Insertion, Deletion, and Traversal).	
8.	Implementation of Circular Queue using Linked List	
UNIT – IV	**TREES**	
9.	Implementation and Traversal (In-order, Pre-order, Post-order) of a Binary Search Tree.	
UNIT – V	**GRAPHS, SORTING AND HASHING**	
10.	Implement Graph Search algorithms (BFS & DFS)	
11.	Implementation of Bubble, Selection and Heap Sorts	
12.	Implementation of Hashing with Collision Resolution with Linear Probing	
Lab Tools and Environments (optional)		

Programming Environment: C-Programming using GCC / VS Code / Code::Blocks	
Core Libraries: Standard C Library	
Database Tools: (optional) None	
Cloud & Collaboration Platforms: (optional) GitHub environment	
Frameworks (basic exposure): (optional) None	
Data Formats & APIs: (optional) None	
TEXT BOOKS & REFERENCES	
1.	Seymour Lipschutz, "Data Structures", McGraw Hill Education; 1st edition, 2014.
2.	R.S. Salaria, "Data Structures", Khanna Publishing House, 2021.
References	
1.	T. H. Cormen, C. E. Leiserson, R L Rivest and C Stein, "Introduction to Algorithms" 3rd Edition, MIT press, 2009.
2.	https://onlinecourses.swayam2.ac.in/e-learning/preview/cec26_cs05
LEARNING OUTCOMES	
1.	Understand fundamental concepts of algorithms, data structures, and their role in problem-solving.
2.	Explain the importance of time-space trade-offs without relying on formal mathematical notations.
3.	Implement and apply linear data structures such as stacks, queues, and linked lists to solve real-world problems.
4.	Navigate hierarchical data structures and understand their role in organized data retrieval.
5.	Use graph structures, sorting, and hashing methods to process and organize data effectively.
PREPARED BY	
Ms.S.E.Viswapriya , Mr.Bhadram Suresh kumar	

Course Code	<<Course Code>>	L	T	P	C
Course Title	FUNDAMENTALS OF MACHINE LEARNING	3	0	4	5

PRE-REQUISITES

Basics of Linear Algebra, Probability, and Statistics
Programming knowledge (Python)
Fundamentals of Data Structures

OBJECTIVES

- To understand the fundamentals and applications of Machine Learning
- To learn supervised and unsupervised learning techniques
- To analyze and evaluate machine learning models
- To apply ML algorithms to real-world datasets
- To introduce advanced topics such as ensemble, reinforcement, and deep learning

COURSE OUTCOMES

At the end of course, the students will be able to

1.	**CO1:** Understand and describe fundamental concepts and applications of machine learning
2.	**CO2:** Apply supervised learning algorithms for prediction tasks
3.	**CO3:** Analyze unsupervised learning techniques for clustering
4.	**CO4:** Evaluate machine learning models using ensemble techniques
5.	**CO5:** Apply reinforcement and deep learning concepts to real-world problems

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	3	2	–	–	–	–	–	–	–	–	–	–	–	–	–
CO 02	3	3	2	–	2	–	–	–	–	–	–	–	–	–	–
CO 03	3	3	2	2	2	–	–	–	–	–	–	–	–	–	–
CO 04	3	3	2	3	2	–	–	–	2	–	–	–	–	–	–
CO 05	2	2	2	2	2	–	–	2	2	2	–	2	–	–	–

LEGEND:1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	INTRODUCTION TO MACHINE LEARNING	9

Definition and characteristics of Machine Learning. Types of Machine Learning: Supervised, Unsupervised, Reinforcement Learning. Applications of Machine Learning. Machine Learning workflow. Dataset handling and preprocessing. Feature selection techniques. Dataset division: Training, Testing, Validation sets. Cross-validation. Overfitting and underfitting.

UNIT – II	SUPERVISED LEARNING	9

Regression and classification concepts. Linear Regression, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM). Model evaluation metrics: MSE, R^2 Score, Confusion Matrix, Precision, Recall, F1-Score, ROC Curve. Bias–variance tradeoff. **Case Study (Local Needs):** Crop yield prediction using regression for Tamil Nadu agricultural data.		
UNIT – III	**UNSUPERVISED LEARNING**	**9**
Introduction to unsupervised learning. Clustering techniques: Hierarchical (Agglomerative, Divisive), K-Means. Cluster evaluation methods. Association rules. Applications. **Case Study (Regional Needs):** Customer segmentation using clustering for regional retail market analysis.		
UNIT – IV	**ENSEMBLE LEARNING**	**9**
Introduction to ensemble learning. Bagging, Boosting. Random Forest. AdaBoost, Gradient Boosting. Voting and stacking. Advantages and limitations. **Case Study (Global Needs):** Recommendation system using ensemble methods for global e-commerce platforms.		
UNIT – V	**REINFORCEMENT LEARNING AND DEEP LEARNING**	**9**
Basics of Reinforcement Learning: agent, environment, reward, policy. Markov Decision Process. Introduction to Deep Learning: Neural Networks, Perceptron, Multilayer Perceptron. Applications in NLP and Computer Vision.		
LIST OF EXPERIMENTS		
UNIT – I	**INTRODUCTION TO MACHINE LEARNING**	
1.	Perform data preprocessing on a given dataset.	
2.	Implement dataset splitting and cross-validation.	
UNIT – II	**SUPERVISED LEARNING**	
3.	Implement Linear Regression and evaluate performance.	
4.	Implement a classification model (KNN/SVM) and compute evaluation metrics.	
UNIT – III	**UNSUPERVISED LEARNING**	
5.	Implement K-Means clustering.	
6.	Perform hierarchical clustering and compare results.	
UNIT – IV	**ENSEMBLE LEARNING**	
7.	Implement Random Forest and analyze performance.	
8.	Implement boosting techniques and compare results.	
UNIT – V	**REINFORCEMENT LEARNING & DEEP LEARNING**	
9.	Implement a basic reinforcement learning model	
10.	Build a neural network using TensorFlow/Keras.	
Lab Tools and Environments (optional)		
Programming Environment: Python, Jupyter Notebook, Google Colab		

Core Libraries: NumPy, Pandas, Matplotlib,Scikit-learn,TensorFlow, Keras	
Database Tools: (optional) MySQL / PostgreSQL – for structured data storage and querying MongoDB – for handling unstructured and semi-structured data SQLite – lightweight database for small ML applications	
Cloud & Collaboration Platforms: (optional) Google Colab – cloud-based Python environment for ML experiments Kaggle – datasets, notebooks, and competitions GitHub – version control and project collaboration AWS / Microsoft Azure / Google Cloud – cloud-based ML deployment (basic exposure)	
Frameworks (basic exposure): (optional) Scikit-learn – core machine learning algorithms TensorFlow – deep learning framework Keras – high-level neural network API PyTorch – emerging deep learning framework	
Data Formats & APIs: (optional) CSV, Excel – structured datasets JSON, XML – semi-structured data formats REST APIs – accessing real-time data Open datasets (Kaggle, UCI Repository	
TEXT BOOKS	
3.	*Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, 3rd Edition, O'Reilly Media, 2022.(Units Covered: I-V)*
4.	*Kevin P. Murphy, Probabilistic Machine Learning: An Introduction, MIT Press, 2022.* ***(Units Covered: I-V)***
5.	*Jeeva Jose, Introduction to Machine Learning, Khanna Book Publishing Co., 2020.* ***(Units Covered: I- V)***
6.	*Rajeev Chopra, Machine Learning, Khanna Book Publishing Co., 2021.* ***(Units Covered: I -IV)***
REFERENCES	
1.	Ethem Alpaydin, *Introduction to Machine Learning*, 4th Edition, MIT Press, 2020.
2.	Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006
3.	Ian Goodfellow, Yoshua Bengio, Aaron Courville, ***Deep Learning***, MIT Press, 2016.
4.	Tom M. Mitchell, *Machine Learning*, McGraw Hill Education, 2017.
5.	John Paul Mueller and Luca Massaron, *Machine Learning for Dummies*, Wiley, 2016.
6	**E-LEARNING RESOURCES** 1. NPTEL / SWAYAM: *Introduction to Machine Learning*, IIT Madras 2. Udacity – *Intro to Machine Learning* 3. Coursera – *Machine Learning (Duke University)*
LEARNING OUTCOMES	
1.	Understand machine learning fundamentals and terminology
2.	Identify and preprocess datasets for analysis
3.	Implement supervised and unsupervised

4.	Evaluate model performance using appropriate metrics
5.	Develop intelligent solutions using ML techniques
PREPARED BY	
Rajalakshmi R Assistant Professor-CSE	

IV – SEMESTER

Course Title	DISCRETE MATHEMATICS	L	T	P	C
Course Code		4	0	0	4

OBJECTIVES

- To familiarize with sets and relations.
- To learn counting techniques
- To familiarize with logic
- To familiarize with algebraic structures and Boolean algebra
- To understand graph networks and its applications

OUTCOMES

For a given logic sentence express it in terms of predicates, quantifiers, and logical connectives. For a given a problem, derive the solution using deductive logic and prove the solution based on logical inference and classify its algebraic structure. Students can evaluate Boolean functions, simplify expressions using the properties of Boolean algebra and develop the given problem as graph networks and solve with techniques of graph theory.

After the successful completion of the course students will be able to

Sl. No	Course Outcome	Bloom's Level
1.	Express a given logical sentence in terms of predicates, quantifiers, and logical connectives	Understanding and Applying
2.	derive the solution for a given a problem using deductive logic and prove the solution based on logical inference and classify its algebraic structure	Understanding and Applying
3.	Evaluate Boolean functions, simplify expressions using the properties of Boolean algebra	Understanding and Applying
4.	Develop the given problem as graph networks and solve with techniques of graph theory.	Understanding and Applying

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12
CO01	S	S	S	S	L	M	S	L	M	M	L	S
CO02	S	S	S	S	S	M	L	L	M	M	S	S
CO03	S	S	M	M	S	M	L	L	M	M	L	S
CO04	S	S	M	M	S	M	L	L	M	M	L	S

UNIT - I **SETS, RELATION AND FUNCTION**

Finite and infinite sets, Countable and uncountable sets, Size of a set, Set operations, Ordered pairs and Cartesian products, Relations, Types of relations, Some operations on relations, Properties of relations, Equivalence classes, Partition of a set, Matrix representation of a relation, Representation of relations by graphs, Hasse diagrams for partial ordering relation.

UNIT - II **COMBINATORICS AND PROOF TECHNIQUES**

Basic counting techniques-Inclusion and Exclusion, Pigeon-hole principle, Permutation and

Combination. Principles of mathematical induction: The Well-Ordering principle, Recursive definition, The Division algorithm: Prime numbers, The Greatest common divisor: Euclidean algorithm.	
UNIT – III	**PROPOSITIONAL LOGIC**
Basic connectives and truth tables, Logical equivalence: The laws of logic, Logical implication, Rules of inference, The use of quantifiers. Proof techniques: Some terminology, Proof methods, and Strategies, Forward proof, Proof by contradiction, Proof by contraposition.	
UNIT – IV	**ALGEBRAIC STRUCTURES AND MORPHISM**
Algebraic structures with one binary operation, Semigroups, Monoids, Groups, Congruence relation and Quotient structures, Permutation groups, Substructures, Normal subgroups, Algebraic structures with two binary operations, Rings, Integral domain, and Fields. Boolean algebra and Boolean ring, Identities of boolean algebra, Duality, Representation of boolean function, Disjunctive and Conjunctive normal form	
UNIT - V	**GRAPHS AND TREES**
Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub graph, Isomorphism, Eulerian and Hamiltonian walks, Graph coloring, Coloring maps, and Planar graphs, Coloring vertices, Coloring edges, List Coloring, Perfect graph, definition properties, and Example, rooted trees, trees and sorting, weighted trees, and prefix codes, Shortest distances by Prim's and Kruskal's algorithm.	
TEXT BOOKS	
1.	Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
2.	Susanna S. Epp, Discrete Mathematics with Applications,4th edition, Wadsworth Publishing Co. Inc.
3.	C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition by, Tata McGraw – Hill.
4.	J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and It's Application to Computer Science", TMG Edition, TataMcgraw-Hill
5.	Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press.
6.	Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson,
7.	Veerarajan, Discrete Mathematics, Tata McGraw – Hill.
PREPARED BY	
Dept of Mathematics	

Course Code		L	T	P	C
Course Title	FUNDAMENTALS OF EMBEDDED SYSTEMS AND IoT	3	0	4	5

PRE-REQUISITES

Basic knowledge in Digital Logics.

OBJECTIVES

- To understand the architecture and programming concepts of microprocessors and microcontrollers for embedded applications.
- To apply Embedded C programming techniques for interfacing memory and I/O devices in embedded systems.
- To analyze real-time operating system concepts and scheduling mechanisms in embedded environments.
- To develop IoT-based applications using Arduino programming with sensor and actuator interfacing.
- To implement IoT communication protocols and open platforms for cloud-connected embedded applications.

COURSE OUTCOMES

At the end of course, the students will be able to

1.	Explain the architecture, instruction set, and programming model of 8086 microprocessor.
2.	Develop assembly language programs and interface peripheral devices using 8051 microcontroller.
3.	Apply Embedded C programming for memory and I/O interfacing with multitasking concepts in embedded systems.
4.	Design Arduino-based IoT applications by integrating sensors and actuators.
5.	Implement IoT communication protocols and open platforms for cloud-enabled embedded system applications.

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	3	2	1	-	1	-	-	-	-	-	-	2	3	2	1
CO 02	3	2	2	-	2	-	-	-	-	-	-	2	3	3	1
CO 03	3	3	2	1	3	-	-	-	1	-	-	2	3	3	2
CO 04	2	2	3	1	3	-	-	-	1	1	-	2	3	3	2
CO 05	2	3	3	2	3	-	-	-	1	1	-	3	3	2	2

LEGEND: 1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	INTRODUCTION TO MICROPROCESSOR (8086)	9

Introduction to 8086 Microprocessor, Architecture, Addressing Modes, Instruction Set, Interrupts and Simple Assembly Language programs: String Operations, Manipulations

UNIT – II	INTRODUCTION TO MICROCONTROLLER (8051)	9

8-Bit Microcontroller – Architecture – Instruction Set and Programming – Programming Parallel Ports – Timers and Serial Port – Interrupt Handling.

UNIT – III	EMBEDDED C PROGRAMMING	9
Memory And I/O Devices Interfacing – Programming Embedded Systems in C – Need For RTOS – Multiple Tasks and Processes – Context Switching – Priority Based Scheduling Policies.		
UNIT – IV	**IoT WITH ARDUINO PROGRAMMING**	**9**
Introduction to Concept of IoT Devices – IoT Configurations – Basic Components – Introduction to Arduino – Types of Arduino – Arduino Tool chain – Arduino Programming Structure – Sketches – Pins – Input/Output From Pins Using Sketches – Introduction to Arduino Shields – Integration of Sensors and Actuators with Arduino		
UNIT – V	**IoT COMMUNICATION AND OPEN PLATFORMS**	**9**
IoT Communication Models and APIs – IoT Communication Protocols – Introduction to Bluetooth, WiFi, ZigBee. Open Platform (Raspberry Pi) – Architecture – Programming – Interfacing – Accessing GPIO Pins – Sending and Receiving Signals Using GPIO Pins .		

LIST OF EXPERIMENTS	
UNIT – I	**INTRODUCTION TO MICROPROCESSOR (8086)**
1.	Write a ALP to perform basic arithmetic operation in various addressing modes on two 16-bit Numbers.
2.	Write a ALP to determine sum of elements in an array (Smallest and Largest).
UNIT – II	**INTRODUCTION TO MICROCONTROLLER (8051)**
3.	Write a microcontroller program to perform basic arithmetic operation on two 8 bit numbers.
4.	Write a ALP to determine sum of elements in an array (Ascending and Descending)
UNIT – III	**EMBEDDED C PROGRAMMING**
5.	Write Basic and arithmetic Programs Using Embedded C.
6.	Develop Embedded C programs to control LEDs/Switches.
UNIT – IV	**IoT WITH ARDUINO PROGRAMMING**
7.	LED Blinking using Arduino Programming.
8.	Interface Push Button with Arduino for Digital Input Control.
UNIT – V	**IoT COMMUNICATION AND OPEN PLATFORMS**
9.	Introduction to Raspberry PI platform.
10.	Interfacing sensors with Raspberry PI.
Lab Tools and Environments (optional)	
Microprocessor (8086) Trainer Kit ,Microcontroller (8051) Trainer Kit, Arduino IDE development board, Raspberry PI kit	

TEXT BOOKS	
1.	Raj Kamal, "Embedded Systems: Architecture, Programming and Design", Tata McGraw-Hill Education, 2nd Edition, 2008, Reprint 2011.
2.	Muhammad Ali Mazidi, Janice Gillispie Mazidi and Rolin D. McKinlay, "The 8051 Microcontroller and Embedded Systems", Pearson Education, 2nd Edition, 2013.
3.	Douglas V. Hall and Andrew L. Rood, "Microprocessors and Interfacing: Programming and Hardware", McGraw-Hill Education, 2nd Edition, 1991.

4.	Arshdeep Bahga and Vijay Madisetti, "Internet of Things: A Hands-on Approach", Universities Press, 1st Edition, 2015.
REFERENCES	
1.	David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things", Cisco Press, 1st Edition, 2017.
2.	Olivier Hersent, David Boswarthick and Omar Elloumi, "The Internet of Things: Key Applications and Protocols", Wiley, 2nd Edition, 2012.
3.	Krishna Kant, "Microprocessors and Microcontrollers: Architecture, Programming and System Design", PHI Learning Pvt. Ltd., 2nd Edition, 2014.
4.	Michael J. Pont, "Embedded C", Pearson Education, 2nd Edition, 2012.
5.	Qing Li and Caroline Yao, "Real-Time Concepts for Embedded Systems", CMP Books, 1st Edition, 2003.
LEARNING OUTCOMES	
By the end of this course, learners will be able to:	
1.	Understand microprocessor and microcontroller architecture and their instruction execution process.
2.	Develop assembly and Embedded C programs for embedded hardware interfacing applications.
3.	Analyze multitasking and scheduling mechanisms in embedded real-time systems.
4.	Design Arduino-based embedded and IoT applications using sensors and actuators.
5.	Implement communication protocols and cloud connectivity in IoT platforms.
PREPARED BY	
Department of ECE & Coordinated by Dr.T.Lakshmibai	

Course Code		L	T	P	C
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	3	0	4	5

PRE-REQUISITES

Basic knowledge of C programming and data structures implementation.

OBJECTIVES

- To understand fundamental concepts of algorithm analysis and asymptotic behavior.
- To apply various algorithm design techniques such as Divide and Conquer, Greedy, and Dynamic Programming.
- To design and analyze graph algorithms for optimization problems.
- To apply state-space search techniques like Backtracking and Branch & Bound.
- To understand Lower Bound Theory and efficient string-matching techniques.

COURSE OUTCOMES

At the end of course, the students will be able to

1.	Analyze the time complexity of iterative algorithms from DS using formal asymptotic notations.
2.	Formulate and solve recurrence relations for recursive algorithms using Master's and Tree methods.
3.	Design and analyze Greedy and Dynamic Programming solutions for real-world optimization.
4.	Apply Backtracking and Branch & Bound to solve "hard" problems through state-space exploration.
5.	Implement string matching and understand the theoretical lower limits of comparison-based algorithms

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	3	3	1	2	–	–	–	1	3	1	1	3	3	1	2
CO 02	3	3	3	2	1	–	–	1	3	3	2	3	3	3	2
CO 03	3	3	2	3	1	–	–	1	3	3	2	3	3	2	3
CO 04	3	3	3	2	1	–	–	1	3	3	2	3	3	3	2
CO 05	3	3	1	2	–	–	–	2	3	2	3	3	3	1	2

LEGEND:1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	FOUNDATIONS OF ALGORITHM ANALYSIS	9

Introduction - Algorithm Characteristics, performance measurements (Time & Space complexity). Total space vs Auxiliary Space. Asymptotic Analysis - Definitions and Properties of best, average, and worst-case behavior. Analysis - Performance analysis of iterative algorithms (Linear Search, Selection Sort, Bubble Sort). Performance Time-Space trade-offs

UNIT – II	RECURSIVE ANALYSIS & DIVIDE AND CONQUER	9

Introduction - Recursion, Sub-Problem reduction, Recursive call stack, base case formulation. Recurrence Relations - Mathematical modeling. Methods for Solving Recurrences - Substitution Method, Recursion Tree Method and Master's Theorem. Divide & Conquer Strategy – Introduction and complexity analysis of Recursive Binary Search, Merge Sort, and quick sort.		
UNIT – III	**GREEDY & DYNAMIC PROGRAMMING**	**9**
Greedy Strategy – Introduction, Fractional Knapsack Problem & analysis, Minimum Spanning Tree definition, Kruskal's and Prim's algorithm, and complexity analysis. Dynamic Programming Strategy - Principle of optimality, Travelling Salesman Problem (TSP) and analysis, Multistage Graph problem & complexity analysis. Shortest Paths – Single Source Shortest Path (Dijkstra's and Floyd-Warshall), All-Pair Shortest Path (Floyd-Warshall) and complexity analysis.		
UNIT – IV	**STATE SPACE & ADVANCED TECHNIQUES**	**9**
State Space Search - Introduction and State Space Tree logic. Backtracking strategy - N-Queen and Graph Coloring problems and complexity analysis. Branch & Bound - 0/1 Knapsack optimization and Traveling Salesperson Problem and complexity analysis. Network Flow – Introduction and Ford-Fulkerson Algorithm (Maximum flow problems).		
UNIT – V	**STRING MATCHING & COMPUTATIONAL LIMITS**	**9**
String Matching - Naive and Brute Force String Matching algorithms. Lower Bound Theory - Theoretical limits of comparison-based sorting. Modern Algorithmic Topics - Introduction to Randomized Algorithms (Las Vegas vs. Monte Carlo); Conceptual analysis of Randomized QuickSort.		
LIST OF EXPERIMENTS		
UNIT – I	**FOUNDATIONS OF ALGORITHM ANALYSIS**	
1.	Implementation of Bubble, Selection Sorts	
UNIT – II	**RECURSIVE ANALYSIS & DIVIDE AND CONQUER**	
2.	Implement Tower of Hanoi & Recursive Binary Search	
UNIT – III	**GREEDY & DYNAMIC PROGRAMMING**	
3.	Implement Fractional Knapsack using greedy method	
4.	Implement Minimum Spanning Tree Kruskal's Algorithm	
5.	Implement Travelling Salesman Problem using Held-Karp	
6.	Implement Dijkstra's algorithm for Single Source Shortest Path	
7.	Implementation of the Floyd-Warshall algorithm for All-Pair Shortest Path	
UNIT – IV	**STATE SPACE & ADVANCED TECHNIQUES**	
8.	Implement Network Flow algorithm using the Ford-Fulkerson method	
9.	Implementation of the N-Queen problem using Backtracking	
UNIT – V	**STRING MATCHING & COMPUTATIONAL LIMITS**	
10.	Implement Randomized algorithm Quick Sort	
11.	Implementation of the Naive String Matching algorithm for pattern detection	
Lab Tools and Environments (optional)		

Programming Environment: C-Programming using GCC / VS Code / Code::Blocks	
Core Libraries: Standard C Libraries	
Database Tools: (optional) None	
Cloud & Collaboration Platforms: (optional) GitHub environment	
Frameworks (basic exposure): (optional) None	
Data Formats & APIs: (optional) None	
TEXT BOOKS & REFERENCES	
7.	Design & Analysis of Algorithms, Gajendra Sharma, Khanna Book Publishing 2018.
8.	Fundamentals of algorithms, Horowitz E, Sahini S, Rajasekaran S., University Press 2008.
References	
1.	Introduction to algorithms, Cormen, Leiserson, Rivest, Stein, 3rd Edition, PHI. 2012
2.	Algorithm Design, Jon Kelinberg and Eva Tardos, 1st Edition, Pearson Education 2014.
3.	Data Structures and Program Design in C By Robert L. Kruse, C.L. Tondo, Bruce Leung, Pearson Education. 2007.
4.	https://onlinecourses.swayam2.ac.in/e-learning/preview/cec26_cs10
LEARNING OUTCOMES	
1.	Mastery in using formal asymptotic notations to quantify the performance of iterative algorithms.
2.	Competence in formulating recurrence relations and applying mathematical methods to solve recursive complexity.
3.	Ability to select and apply Greedy and Dynamic Programming strategies to solve complex engineering optimization problems.
4.	Understanding of state-space exploration through Backtracking, Branch & Bound, and Network Flow techniques.
5.	Knowledge of theoretical lower bounds in computation, string matching logic, and modern randomized algorithmic approaches.
PREPARED BY	
Ms.S.E.Viswapriya , Mr.Bhadram Suresh kumar	

Course Code		L	T	P	C
Course Title	OBJECT ORIENTED PROGRAMMING	3	0	4	5

PRE-REQUISITES

Basic Programming skills.

OBJECTIVES

•	To understand the concept and principles of OOP as well as the purpose and usage of Exception Handling
•	Design and implement a Collection classes and integration with real time applications
•	To develop programs using the Java Collection API as well as the Java standard class library.
•	To learn and understand Packages, Multi-threading concepts
•	To Learn and Understand the concept of I/O, Applet and GUI programming with AWT

COURSE OUTCOMES

At the end of course, the students will be able to

1.	Apply the concepts and Purpose of Object oriented Programming concepts for developing and Implementing required software.
2.	Apply the principles of Collections classes and file concepts to implements various software applications
3.	Implement the applications by using java API concepts, to discover errors of Java programs for collaborative programming/editing.
4.	Design and Implementation of creating and using packages and Applet.
5.	Design and Implements the concept of GUI based software applications using appropriate GUI API.

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	3	2	2	1	2	-	-	-	-	-	-	1	3	2	1
CO 02	2	2	3	2	3	-	-	-	-	-	-	1	3	2	1
CO 03	2	3	2	2	3	-	-	-	1	-	-	1	2	3	1
CO 04	2	2	3	2	3	-	-	-	-	1	-	1	3	3	2
CO 05	2	2	3	1	3	-	-	-	1	1	-	1	3	2	2

LEGEND: 1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	**INTRODUCTION TO OOPS**	**9**

Overview of OOP – Object oriented programming paradigms, Concepts of object-oriented programming- C++ programming basics- Data types, Manipulators, Cin, Cout, Type conversion – Class and objects: Basics of class and objects-Overview of Java-Java Vs C++

UNIT – II	**JAVA BASICS**	**9**

JVM- Java Basics: Java Tokens, Programming Structures in Java – Defining classes in Java –

Constructors Methods -Access specifiers – Static members- Java Doc comments-String Handling: String Basics, String Operations.		
UNIT – III	**INHERITANCE AND INTERFACES**	**9**
Inheritance: Inheritance Syntax and types – Multiple Inheritance - Super keyword - Overloading Methods – Objects as Parameters-Method Overriding–Abstract Classes-Interfaces: Defining Interfaces, Extending Interfaces, Implementing Interfaces, Accessing Interfaces.		
UNIT – IV	**PACKAGES, EXCEPTION HANDLING AND MULTITHREADING**	**9**
Packages: Built-in packages- Creating, Accessing and Using Packages, Hiding Classes, Import command. Exception Handling basics – Multiple catch– Nested try – Java's Built-in Exceptions – User defined Exception. Multithreaded Programming: Java Thread Model–Creating a Thread and Multiple Threads – Priorities –Multithreading life cycle.		
UNIT – V	**I/O ,COLLECTIONS, APPLET AND AWT**	**9**
I/O Basics – Reading and Writing Console I/O – Reading and Writing Files. Exploring java.Util: collections- Applet Fundamentals-Java Application Vs Java Applets, Applet life Cycle, AWT (Abstract windows toolkits), components, Containers.		
LIST OF EXPERIMENTS		
UNIT – I	**INTRODUCTION TO OOPS**	
1.	C++ Basics: Create a program to calculate employee gross salary using cin, cout, and manipulators for formatted output.	
2.	Type Conversion: Implement a program to demonstrate explicit and implicit type casting between different data types in C++.	
3.	Class Foundations: Design a class with data members and member functions to input and display data (use Access Specifiers also).	
UNIT – II	**JAVA BASICS**	
4.	Constructor Overloading: Create a Box class with multiple constructors to initialize dimensions (cube, rectangle, or default).	
5.	Static Members: Write a program to count the number of objects created for a class using a static variable and a static method.	
6.	String Operations: Develop a Java program by applying sting functions.	
UNIT – III	**INHERITANCE AND INTERFACES**	
7.	Implement Multilevel Inheritance	
8.	Implement Method Overriding in JAVA	
9.	Write JAVA program to implement Multiple Inheritance.	
UNIT – IV	**PACKAGES, EXCEPTION HANDLING AND MULTITHREADING**	
10.	User-defined Packages: Create a package com.math.operations containing a class for basic arithmetic and import it into a main application.	
11.	Exception Handling: Write a program that handles ArithmeticException (division by zero) and ArrayIndexOutOfBoundsException using a nested try-catch block.	
12.	Write JAVA program to implement Multithreading concept.	
UNIT – V	**I/O ,COLLECTIONS, APPLET AND AWT**	

13.	File I/O: Write a Java program that perform all file related operations
14.	Implement Collection Framework.
15.	AWT/Applet Application: Design a simple "Calculator" GUI using AWT components (Buttons, TextFields, Labels).
Lab Tools and Environments (optional)	
Programming Environment: JDK Latest version	
TEXT BOOKS	
1.	Herbert Schildt, "Java The Complete Reference", 13th Edition, Tata McGraw Hill, 2025.
2.	JAVA : A Beginner's Guide Tenth Edition November 2024
3.	James Jaworski, "Java Unleashed", 4th revised edition, SAMS Tech media Publications, Digitized-2010.
REFERENCES	
1.	Kogent Solution Inc ,Java 6 Programming Black Book, New Ed, Dream tech Press, 2007
2.	Campione, Walrath and Huml, "The Java Tutorial", Addison Wesley, 2001.
3.	Elliotte Rusty Harold ,Java Network Programming, fourth Edition, O'Reilly Media, Inc.", 2013,
4.	Java Database Programming Bible, John O' Donahue, illustrated Edition, Wiley, 2002 (for Unit -V)
5.	Fundamentals of Java Programming , Authors: Ogihara, Mitsunori (2018)- ISBN 978-3-319-89491-1
WEB SOURCE REFERENCE	
https://onlinecourses.nptel.ac.in/noc22_cs47/preview	
LEARNING OUTCOMES	
By the end of this course, learners will be able to:	
1.	Students will be able to design and develop modular, reusable software components by applying core Object-Oriented Programming (OOP) principles such as inheritance, polymorphism, and encapsulation.
2.	Learners will be able to implement robust error-handling mechanisms using Java's Exception Handling framework
3.	Students will be able to create concurrent applications using Multithreading.
4.	Students will be able to utilize the Java Collection API and File I/O operations to efficiently store, retrieve, and manage complex data structures in real-time applications.
5.	Students will be able to design and implement interactive, user-friendly Graphical User Interfaces (GUIs) by leveraging AWT.
PREPARED BY	
Dr.P.Shanmugapriya and Dr.M.Saraswathi	

Course Code		L	T	P	C
Course Title	ORGANIZATIONAL BEHAVIOUR	3	0	0	3

PRE-REQUISITES

OBJECTIVES

•	To provide a comprehensive understanding of the fundamental concepts, evolution, and significance of Organizational Behaviour in modern management contexts.
•	To develop insights into individual behaviour by examining perception, learning processes, and their influence on workplace effectiveness.
•	To analyze personality traits, theories, and individual differences, including their implications for employee behaviour and organizational outcomes.
•	To evaluate motivational theories, attitudes, and job satisfaction factors to enhance employee performance and organizational productivity.
•	To understand group behaviour and team dynamics, enabling effective management of teams, collaboration, and workplace challenges.

COURSE OUTCOMES

At the end of course, the students will be able to

1.	Understand the fundamental concepts, evolution, and significance of Organizational Behaviour in contemporary organizations. *(K2)*
2.	Examine the role of perception and learning processes in shaping individual behaviour at the workplace. *(K4)*
3.	Evaluate personality traits, theories, and individual differences and their impact on employee behaviour and organizational effectiveness. *(K5)*
4.	Apply motivational theories and attitude-related concepts to improve job satisfaction and employee performance. *(K3)*
5.	Analyze group behaviour and team dynamics to enhance collaboration and manage workplace challenges effectively. *(K4)*

POs and COs MAPPING TABLES

	PO 01	PO 02	PO 03	PO 04	PO 05	PO 06	PO 07	PO 08	PO 09	PO 10	PO 11	PO 12	PSO 01	PSO 02	PSO 03
CO 01	1	2	–	–	–	–	–	–	–	2	–	1	–	–	–
CO 02	–	3	–	–	–	–	–	–	–	2	–	–	–	1	–
CO 03	–	2	–	–	–	–	–	–	1	2	–	–	–	2	–
CO 04	–	2	–	–	–	–	–	–	1	–	1	–	–	2	1
CO 05	–	2	–	–	–	–	–	–	3	2	1	–	–	2	1

LEGEND:1-LOW, 2 – MEDIUM, 3-HIGH

UNIT – I	**INTRODUCTION TO ORGANIZATIONAL BEHAVIOUR**	**9**

Meaning and Definition of Organizational Behaviour –Contributing disciplines of OB – Roles of OB in Contemporary Management – Historical Evolution of OB – OB as a Science– Emerging Issues in Organizational Behaviour

UNIT – II	**PERCEPTION AND LEARNING**	**9**
Perception: Meaning, Elements, and Perceptual Process – Perceptual Selection, Grouping, and Biases – Learning: Social Learning Theory and Role Modelling		
UNIT – III	**PERSONALITY**	**9**
Personality: Meaning, Theories, and Assessment – Dark Triad and Abusive Personality – Personality Disorders and Workplace Implications		
UNIT – IV	**MOTIVATION AND ATTITUDES**	**9**
Motivation: Meaning and Process – Types of Motives – Theories of Motivation – Aligning Motivation with Performance – Incentives and Reward Systems – Work Motivation in Indian Context -Attitudes: Growth vs Fixed Mindset -Job Satisfaction: Causes and Consequences – Techniques for Improving Job Satisfaction.		
UNIT – V	**GROUP BEHAVIOUR AND TEAM DYNAMICS**	**9**
Nature of Groups and Teams – Stages of Group Development – Team Roles and Dynamics – Group Cohesiveness – Managing Team Performance – Problems in Teamwork (Free Riding, Social Loafing, Groupthink) – Virtual Teams and Team Analytics		
TEXT BOOKS		
1.	Stephen P. Robbins, Timothy A. Judge, Organisational Behavior, PHI Learning / Pearson Education, 16th edition, 2014	
2.	Fred Luthans, Organisational Behavior, McGraw Hill, 12th Edition, 2013.	
3.	Understanding Organizational Behaviour by Udai Pareek, Oxford University Press (Third Edition).	
4.	Behaviour in Organizations by Jerald Greenberg and Robert A. Baron, PHI learning private Ltd, New Delhi (Ninth Edition).	
5.	Ivancevich, Konopaske&Maheson, OranisationalBehavior& Management, 7th edition, Tata McGraw Hill, 2008.	
REFERENCES		
1.	McShane, Mary V. Glinow, Organizational Behavior, 8th Edition, Tata Mc Graw Hill, 2017.	
2.	Nelson, Quick, Khandelwal. ORGB – An innovative approach to learning and teaching. Cengage learning. 2nd edition. 2012	
LEARNING OUTCOMES		
1.	Describe the fundamental concepts, evolution, and significance of Organizational Behaviour in modern organizations.	
2.	Explain the role of perception and learning processes in influencing individual behaviour at the workplace.	
3.	Examine personality traits, theories, and their implications for employee behaviour and organizational effectiveness.	
4.	Apply motivation theories and attitude-related concepts to enhance job satisfaction and employee performance.	
5.	Analyze group behaviour and team dynamics to improve collaboration and effectiveness in organizational settings.	
PREPARED BY		
Dept. of MBA		

Course Code		L	T	P	C
Course Title	INDIAN KNOWLEDGE SYSTEM	1	-	-	NC

UNIT – I	**History of Indian Knowledge System**	
Genesis of Bhartiya Knowledge System-History of IKS		
UNIT – II	**India's characteristic knowledge & India's epistemology**	
IKS: Nature, Philosophy and Character-India's Epistemology- Knowledge Frameworks & Classification		
UNIT – III	**Ancient Education System**	
Ancient Scriptures-Ancient Education-Educating Sciences- Khagol Vijnana (Astronomy)-Vastukala (Architecture)- Ayurveda-Krishi Vijnana (Agricultural) Practices		
UNIT – IV	**Scientific approaches of IKS & Torch-bearers**	
Dhatu Vijnana (Metallurgy)- Ganita: Mathematics in India-Yuddha Vidhya (Military Sciences)- Niyuddha Kala (Martial Arts)- Environmental Sciences		
UNIT – V	**Scientific approaches of IKS & Torch-bearers**	
Chandashastra (Prosody)-Bhasa Va Vyakarana (Language and Grammar)- Bharata's Natyashastra (Science of Drama, Dance and Music)		

TEXT BOOKS

1.	Introduction to Indian Knowledge System: Concepts and Applications, Archak, K.B. (2012). Kaveri Books, New Delhi.ISBN-13:978-9391818203
2.	Introduction To Indian Knowledge System: Concepts and Applications, Mahadevan, B.Bhat, Vinayak Rajat,Nagendra Pavana R.N.PHI, ISBN: 9789391818203
3.	Glimpse into Kautilya's Arthashastra Ramachandrudu P. (2010), Sanskrit Academy, Hyderabad ISBN:9788380171074
4.	"Introduction" in Studies in Epics and Purāṇas, (Eds.), KM Munshi and N Chandrashekara Aiyer Bhartiya Vidya Bhavan

REFERENCES

1.	*https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr26_ed18*