LINGAYA’S VIDYAPEETH

Department of Computer Science & Engineering

Scheme & Syllabus

Of

B.Tech.(CSE)

Batch: 2024-2028

VISION

To be a school, committed to education, research & innovation and develop globally competent professionals in the area of Computer Science, Information Technology and Computer Applications who are responsible citizens and have respect for life and sensitivity towards environment.

MISSION

1. To develop professionals and leaders in Computer Science, IT and allied areas who have right attitude and aptitude to serve the society.
2. To develop and maintain state-of-the-art infrastructure and research facilities to enable create, apply and disseminate knowledge.
3. To foster linkages with all stakeholders for continuous improvement in academics in Computer Science, IT and Computer Applications.
4. To develop human potential to its fullest extent so that intellectually capable and imaginatively gifted leaders can emerge who have deep respect for human life and values.
5. To undertake disciplinary and inter-disciplinary collaborative research and innovation which offer opportunities for long term interaction with academia and industry and develop technologies relevant to the society.

PROGRAM OUTCOMES:

PO1- Engineering Knowledge: Apply the knowledge of mathematics, science, engineering and Application fundamentals, and an engineering and Application specialization to the solution of complex engineering problems.

PO2- Problem Analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3-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.

PO4- 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.

PO5- 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.

PO6- 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.

PO7- 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.

PO8- Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9- Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10- 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.

PO11- 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.

PO12- Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES:

PSO1: To equip the students with theoretical and implementation knowledgebase in all the latest areas of Computer Science & Engineering for a successful career in software industries, pursuing higher studies, or entrepreneurial establishments.

PSO2: To nurture the students with the critical thinking abilities for better decision making by offering them a socially acceptable solutions to real life problems through computing paradigm.

PSO3: To nurture the students with the comprehensive analytical and design abilities by offering them techno-commercially feasible solutions of real business problems through computing.

LINGAYA'S VIDYAPEETH

SCHEME OF STUDIES

SESSION: 2024-25

School: School of Engineering & Technology	**Batch: 2024-28**
Department: CSE	**Year:1st**
Course: B.Tech (CSE)	**Semester: I**

SN	**Cate-gory**	**Course Code**	**Course Name**	**Periods**			**Credits**	**Evaluation Scheme**					**Subject Total Marks**
								Theory			**Practical**		
				L	**T**	**P**		**ABQ**	**MSE**	**ESE**	**IP**	**EXP**	
1	BSC	BS-107E	Engineering Mathematics-I	3	1	0	4	15	25	60	-	-	100
2	BSC	BS-109E	Engineering Physics & Chemistry	3	0	2	4	15	25	60	60	40	200
3	ESC	EC-101E	Integrated Electrical and Electronics theory & practice	3	0	2	4	15	25	60	60	40	200
4	ESC	CS-101E	Problem Solving using C & C++	3	0	0	3	15	25	60	-	-	100
5	HSMC	HSS-101	Effective Technical Communication-I	2	0	4	4	15	25	60	60	40	200
6	ESC	ME- 151E	IDEA Lab Workshop	0	0	4	2	-	-	-	60	40	100
7	HSMC	HSS-103E	Universal Human Values	1	0	0	1	40		60	-	-	100
8	ESC	CS-151E	Problem Solving using C & C++ Lab	0	0	2	1	-	-		60	40	100
9	PROJ	PROJ-151E	Innovative projects 1.0 (Phase-I)/IIC	0	0	4	2	-	-	-	60	40	100
10	AU	AU- 101E	Yoga & Practice/NSS	2	0	0	0	-	-	-	-	-	-
11	AU	HSS-151	Extra & co-curricular	0	0	2	0	-	-	-	-	-	-
			Total	**17**	**1**	**20**	**25**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
PROJ:	Project	ESE:	End Semester Examination
PDP:	Personality Development Programme	IP:	Internal Practical
L:	Lecture	EXP:	External Practical

LINGAYA'S VIDYAPEETH
SCHEME OF STUDIES
SESSION: 2024-25

School: School of Engineering & Technology	**Batch: 2024-28**
Department: CSE	**Year:1st**
Course: B.Tech (CSE)	**Semester: II**

SN	Cate-gory	Course Code	Course Name	Periods			Credits	Evaluation Scheme					Subject Total Marks
								Theory			Practical		
				L	T	P		ABQ	MSE	ESE	IP	EXP	
1	BSC	BS-108E	Engineering Mathematics-II	3	0	0	3	15	25	60	-	-	100
2	PCC	CS-102E	Python Programming	3	0	0	3	15	25	60			100
3	PCC	CS-104E	Design thinking with AI	3	0	0	3	15	25	60	-	-	100
4	BSC	BS-110E	Integrated Environment Science & Engineering theory	2	0	2	3	15	25	60	-	-	100
5	HSMC	HSS-102E	Effective Technical Communication-II	2	0	4	4	15	25	60	60	40	100
6	ESC	ME-152E	Digital Fabrication & manufacturing practices	0	0	2	1	-	-	-	60	40	100
7	ESC	ME-154E	Engineering Graphics Practices	0	0	4	2	-	-	-	60	40	100
8	PCC	CS-152E	Python Programming lab	0	0	2	1	-	-	-	60	40	100
9	PEC	PEC(EC/CS/ME/CE)-102E	MOOC courses (NPTEL)	2	0	0	2	-	-	-	-	-	100
10	PROJ	PROJ-152E	Innovative projects2.0 (phase-II)/IIC	0	0	4	2	-	-	-	60	40	100
11	AU	AU-152E	Introduction to Soft skills	0	0	2	0	-	-	-	-	-	-
			Total	**15**	**0**	**20**	**24**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
PROJ:	Project	ESE:	End Semester Examination
PDP:	Personality Development Programme	IP:	Internal Practical
L:	Lecture	EXP:	External Practical
T:	Tutorial		
P:	Practical		

LINGAYA'S
VIDYAPEETHSCHEME OF
STUDIES SESSION: 2025-26

School: School of Engineering & Technology	**Batch:2024-28**
Department: CSE	**Year:2nd**
Course: B.Tech (CSE)	**Semester: III**

S N	**Cate-gory**	**Course Code**	**Course Name**	**Periods**			**Cred its**	**Evaluation Scheme**					**Subjec tTotal Marks**
								Theory			**Practical**		
				L	**T**	**P**		**ABQ**	**MSE**	**ESE**	**IP**	**EXP**	
1	PEC	EC-203E	Digital Electronics	3	0	0	3	15	25	30	-	-	100
2	PCC	CS-203E	Computer Architecture and Organization	3	0	0	3	15	25	30	-	-	100
3	PCC	CS-205E	Discrete Mathematics & Graph Theory	3	0	0	3	15	25	30	-	-	100
4	PCC	CS-207E	Operating System	3	0	0	3	15	25	30	-	-	100
5	PCC	CS-209E	Data Structure Using C++	3	0	0	3	15	25	30	-	-	100
6	MC	BS-203E	Numerical & statistical Methods	3	1	0	4	15	25	30	-	-	100
7	PEC	EC-253E	Digital Electronics Lab	0	0	2	1	-	-	-	60	40	100
8	PCC	CS-257E	Operating System Lab	0	0	2	1	-	-	-	60	40	100
9	PCC	CS-259E	Data Structure Using C++ Lab	0	0	2	1	-	-	-	60	40	100
10	MC	PEP-201	Soft Skills - I	1	0	2	2	-	-	-	60	40	100
			Total	**19**	**1**	**8**	**24**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
PROJ:	Project	ESE:	End Semester Examination
PDP:	Personality Development Programme	IP:	Internal Practical
L:	Lecture	EXP:	External Practical

LINGAYA'S
VIDYAPEETHSCHEME OF
STUDIES SESSION: 2025-26

School: School of Engineering & Technology	**Batch: 2024-28**
Department: CSE	**Year:2nd**
Course: B.Tech (CSE)	**Semester: IV**

SN	**Cate- gory**	**Course Code**	**Course Name**	**Periods**			**Credi t s**	**Evaluation Scheme**					**Subjec tTotal Marks**
								Theory			**Practical**		
				L	**T**	**P**		**A B Q**	**MS E**	**ES E**	**IP**	**EX P**	
1	PCC	CS-202E	Database Management System	3	0	0	3	15	25	60	-	-	100
2	PCC	CS-204E	Design and Analysis of Algorithm	3	0	0	3	15	25	60	-	-	100
3	PCC	CS-206E	Object Oriented programming using Java	3	0	0	3	15	25	60	-	-	100
4	PCC	CS-208E	Artificial Intelligence	3	0	0	3	15	25	60	-	-	100
5	PCC	CS-210E	Computer Network	3	0	0	3	15	25	60	-	-	100
6	PCC	CS-212E	Theory of Computation	3	0	0	3	15	25	60	-	-	100
7	MC	MC-202	Indian Constitution	1	0	0	0	-	-	-	-	-	100
8	MC	CS-216E	Basic & Advanced Excel	0	0	4	2	-	-	-	60	40	100
9	PCC	CS-252E	Database Management System Lab	0	0	2	1	-	-	-	60	40	100
10	PCC	CS-256E	Object Oriented programming using Java Lab	0	0	2	1	-	-	-	60	40	100
11	PCC	CS-258E	Artificial Intelligence Lab	0	0	2	1	-	-	-	60	40	100
12	AUC	VAC-202E	Value Added Course-I	0	0	0	0	-	-	-	-	-	50
			Total	**19**	**0**	**10**	**23**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
PROJ:	Project	ESE:	End Semester Examination
PDP:	Personality Development Programme	IP:	Internal Practical

LINGAYA'S VIDYAPEETHSCHEME OF STUDIES SESSION: 2026-27

School: School of Engineering & Technology	**Batch: 2024-28**
Department: CSE	**Year:3rd**
Course: B.Tech (CSE)	**Semester: V**

SN	Cate-gory	Course Code	Course Name	Periods			Credits	Evaluation Scheme					SubjectTotal Marks
								Theory			Practical		
				L	T	P		ABQ	MSE	ESE	IP	EXP	
1	PCC	CS-301E	Machine Learning	3	0	0	3	15	25	60	-	-	100
2	PCC	EC-301E	Microprocessor & Microcontroller	3	0	0	3	15	25	60	-	-	100
3	PCC	CS-303E	Compiler Design	3	1	0	4	15	25	60	-	-	100
4	PCC	CS-305E	Agile Development & Methodology	3	0	0	3	15	25	60	-	-	100
5	PCC	CS-307E	Web Technology & Services	3	0	0	3	15	25	60	-	-	100
6	MC	RP-301E	Literature Approach & Scientific Writing	3	0	2	4	-	-	-	-	-	100
7	HSMC	PEP-301	Leadership & Management Skills	1	0	2	2	-	-	-	-	-	100
8	PCC	CS-351E	Machine Learning Lab	0	0	2	1	-	-	-	60	40	100
9	PCC	EC-351E	Microprocessor & Microcontroller Lab	0	0	2	1	-	-	-	60	40	100
10	PCC	CS-357E	Web Technology & Services Lab	0	0	2	1	-	-	-	60	40	100
11	PROJ	PROJ-351E	Minor Project - I	0	0	2	1	-	-	-	-	100	100
			Total	**19**	**1**	**12**	**26**						

LINGAYA'S VIDYAPEETHSCHEME OF STUDIES SESSION: 2026-27

School: School of Engineering & Technology										**Batch: 2024-28**			
Department: CSE										**Year:3rd**			
Course: B.Tech (CSE)										**Semester: VI**			
SN	**Cate gory**	**Course Code**	**Course Name**	**Periods**			**Cre dits**	**Evaluation Scheme**					**Subject Total Marks**
								Theory			**Practical**		
				L	**T**	**P**		**ABQ**	**MS E**	**ES E**	**IP**	**EXP**	
1	PCC	CS-302E	Software Engineering	3	0	0	3	15	25	60	-	-	100
2	PCC	CS-304E	Big Data Analytics	3	0	0	3	15	25	60	-	-	100
3	PCC	ABMA_CS	Academic Basket (Preference-I)	4	0	0	4	15	25	60	-	-	100
4	PCC	ABMI_CS	Academic Basket (Preference-II)	2	0	0	2	15	25	60	-	-	100
5	PCC	CS-306E	Business Intelligence	3	0	0	3	15	25	60	-	-	100
6	PEC		Elective – I	3	0	0	3	15	25	60	-	-	100
7	MC	MC-302	Essence of Indian traditional Knowledge	1	0	0	0	-	-	-	-	-	100
8	PCC	CS-352E	Software Engineering Lab	0	0	2	1	-	-	-	60	40	100
9	PCC	CS-358E	Mongo DB	0	0	4	2	-	-	-	60	40	100
10	PROJ	PROJ-352E	Minor Project -II	0	0	2	1	-	-	-	-	100	100
11	PEC	PEC(CS)-302E	MOOC Course – II (NPTEL)	1	0	0	3	-	-	-	-	-	100
12	AUC	VAC-302E	Value Added Course-II	0	0	0	0	-	-	-	-	-	50
			Total	**20**	**0**	**8**	**25**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
PROJ:	Project	ESE:	End Semester Examination
PDP:	Personality Development Programme	IP:	Internal Practical
L:	Lecture	EXP:	External Practical

LINGAYA'S VIDYAPEETHSCHEME OF STUDIES SESSION: 2027-28

School: School of Engineering & Technology	**Batch: 2024-28**
Department: CSE	**Year:4th**
Course: B.Tech (CSE)	**Semester: VII**

S N	**Cate-gory**	**Course Code**	**Course Name**	**Periods**			**Cr edits**	**Evaluation Scheme**					**Subjec tTotal Marks**
								Theory			**Practical**		
				L	**T**	**P**		**A B Q**	**MS E**	**ES E**	**IP**	**EX P**	
1	PCC	CS-401E	Deep Learning	3	0	0	3	15	25	60	-	-	100
2	PCC	ABMA_C S	Academic Basket (Preference-I)	4	0	0	4	15	25	60	-	-	100
3	PCC	ABMI_CS	Academic Basket (Preference-II)	2	0	0	2	15	25	60	-	-	100
4	PEC		Elective – II	3	0	0	3	15	25	60	-	-	100
5	PCC	CS-403E	R Programming	3	0	0	3	15	25	60	-	-	100
6	PCC	PEP-401	Professional Skills	1	0	2	2	-	-	-	-	-	100
7	RD	RD-401	IPR	2	0	0	2	-	-	-	-	-	100
8	PCC	CS-451E	Deep Learning lab	0	0	2	1	-	-	-	60	40	100
9	PCC	CS-453E	R Programming Lab	0	0	2	1	-	-	-	60	40	100
10	PROJ	PROJ-451E	Major Project (Phase-I)	0	0	8	4	-	-	-	-	100	100
			Total	**18**	**0**	**14**	**25**						

LINGAYA'S VIDYAPEETHSCHEME OF STUDIES SESSION: 2027-28

School: School of Engineering & Technology	**Batch: 2024-28**
Department: CSE	**Year:4th**
Course: B.Tech (CSE)	**Semester: VIII**

SN	Cate-gory	Course Code	Course Name	Periods			Credits	Evaluation Scheme					Subject Total Marks
								Theory			Practical		
				L	T	P		ABQ	MSE	ESE	IP	EXP	
1	PROJ	PROJ-452E	Internship/ Major Project (Phase- II)	0	0	30	15	-	-	-	-	100	100
2	PRC	CS-484E	Seminar	0	0	4	2	-	-	-	-	100	100
3	PEC	PEC(CS)-402E	MOOC Course –III(NPTEL)	1	0	0	3	-	-	-	-	-	100
			Total	**1**	**0**	**34**	**20**						

Abbreviations:

PCC:	Programme Core Courses	ABQ:	Assignment Based Quiz
PEC:	Programme Elective Courses	MSE:	Mid Semester Examination
PROJ:	Project	ESE:	End Semester Examination
PDP:	Personality Development Programme	IP:	Internal Practical
L:	Lecture	EXP:	External Practical
T:	Tutorial	PSC:	Program Skill Course
P:	Practical		

LINGAYA'S VIDYAPEETH
SCHEME OF STUDIES

Departmental Elective

	Code	Subject
Elective-I	CSE-302E	Cryptography & Data Compression
	CSE-304E	Data Mining & Data warehousing
	CSE-306E	Neural Network
	CSE-308E	Predictive Analysis & Modeling
	CSE-310E	Cloud Computing
Elective-II	CSE-401E	Image Processing
	CSE-403E	Social Network Analytics
	CSE-405E	Data Science
	CSE-407E	IOT

Syllabus

of

1st & 2nd Semester

CS-101E	PROBLEM SOLVING USING C & C++	L-T-P	Cr
		3-0-0	3

OBJECTIVE

Understand Core Programming Concepts: Grasp fundamental programming constructs such as variables, loops, and conditionals. Learn Structured and Procedural Programming: Develop the ability to write clear, logical, and efficient code using structured programming techniques in C. Master Object-Oriented Programming (OOP): Apply OOP principles like encapsulation, inheritance, and polymorphism in C++.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Know the basic concepts of programming languages as well as operating system

CO2: Learn the basics of programming using C

CO3: Undergo the functions and pointers

CO4: Learn about the structures, unions as well as functions using recursion

CO5: Know about the dynamic programming as well as file handling

UNIT I

BASICS OF C PROGRAMMING: Problem definition, Representation of Algorithms: Flow charts/ Pseudocode with example, Introduction to C: Structure of C program, C character set, Identifier and Keywords, constants, variables, Data types ,Type conversion, Types of operators, Input and output functions in C, header files, common programming errors, Control Statements, Sequencing, Condition and iteration.

Arrays and Strings: Declaring, Referencing and initializing arrays, array subscript, using for loop for sequential access, multi-dimensional array, String basics string library functions, assignment and substring, concatenation, string comparison, etc.

UNIT II

FUNCTIONS AND POINTERS: Definition of function, function prototype, Purpose of main function, passing parameters, Scope of function, recursion, Call by value and reference, Types of storage classes, Scope of variable: Global and local, static variables, Recursion. Pointer variables, initializing pointers, pointer operators, pointer expressions, pointers and arrays, pointer and functions.

UNIT III

STRUCTURES & UNIONS: Defining a structure, Declaring structure variables, Structure initialization, Copying and Comparing Structure variables, Array of structures, Arrays within structure, nested structures, Unions. Recursion as a different way of solving problems.

UNIT IV

DYNAMIC ALLOCATION, AND FILE HANDLING: C's dynamic allocation functions. Streams and file types, opening and closing a data file, input and output operations, text mode versus binary mode, formatted input output operations with files, random access to files.

UNIT V

OBJECT ORIENTED PROGRAMMING CONCEPTS:

Introduction: Procedural Programming Vs Object Oriented Programming, OOP concepts- Class, Object, Abstraction, Encapsulation, Inheritance, Polymorphism, Function Overloading, Operator Overloading, Constructors, Virtual function.

TEXT/REFERENCE BOOKs

1. Byron S. Gottfried (1996), Programming with C, McGraw-Hill Education and ISBN: 9780070240353.
2. E. Balagurusamy (1990), Programming in C, Tata McGraw-Hill Publishing ISBN: 9780074600474.
3. E. Balagurusamy, Object Oriented Programming with C++, Tata McGraw-Hill Publishing ISBN: 9389949181.

CS-151E	PROBLEM SOLVING USING C & C++ LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

Understand the basics of C and C++ programming languages. Learn to write efficient and structured programs using C and C++. Develop problem-solving skills through algorithms and flowcharts. Master concepts like data types, loops, functions, arrays, and pointers. Implement object-oriented programming (OOP) concepts in C++ (classes, inheritance, polymorphism).

COURSE OUTCOMES

CO1 Read, understand and trace the execution of programs written in C language.

CO2 Write the C code for a given algorithm.

CO3 Implement Programs with pointers and arrays, perform pointer arithmetic, and use the pre-processor.

CO4 Write programs that perform operations using derived data types.

Program List

SEQUENTIAL CONTROL STATEMENTS

1. Write a program to calculate simple interest
2. Write a program to calculate average of three numbers
3. Write a program to add digits of a four digit number
4. Write a program to check whether the person if eligible for voting or not

CONDITIONAL CONTROL STATEMENTS

5. Write a program to find greatest of two numbers
6. Write a program to find out which type of triangle it is
7. Write a program to find out greatest of three numbers
8. Write a program to evaluate performance of the student
9. Write a program to make a basic calculator

LOOP CONTROL STATEMENTS

10. Write a program to print Fibonacci up-to the given limit
11. Write a program to find the sum of digits of a number
12. Write a program to find factorial of a number
13. Write a program to print table of any number
14. Write program for printing different pyramid pattern

ARRAYS AND STRINGS

15. Write a program to find the sum and average of 50 students
16. Write a program to sort the array elements
17. Write a program to add 2 matrix

18. Write a program to multiply 2 matrices
19. Write a program to perform sting operations

FUNCTIONS & POINTERS

20. Write a program to calculate factorial of a number using recursive function
21. Write a program to print first n Fibonacci using recursive function
22. Write a program using function to swap two numbers using call by reference

STRUCTURES

23. Write a program to read an employee record using structure and print it
24. Write a program to prepare salary chart of employee using array of structures

FILE HANDLING

25. Write a program to count the number of characters, spaces, tabs, new line characters in a file.
26. Write a program to receive strings from keyboard and write them to a file.
27. Write a program to copy a file to another.
28. Write a program to read strings from a file and display them on screen

OBJECT ORIENTED PROGRAMMING USING C++ PROGRAM LIST

29. Write a program that uses a class where the member functions are defined inside a class.
30. Write a program that uses a class where the member functions are defined outside a class.
31. Write a program to demonstrate the use of static data members.
32. Write a program to demonstrate the use of const data members.
33. Write a program to demonstrate the use of zero argument and parameterized constructors.
34. Write a program to demonstrate the use of dynamic constructor..
35. Write a program to demonstrate the use of explicit constructor..
36. Write a program to demonstrate the use of initializer list.
37. Write a program to demonstrate the overloading of increment and decrement operators.
38. Write a program to demonstrate the multilevel inheritance..
39. Write a program to demonstrate the exception handling.

CS-102E	PYTHON PROGRAMMING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To build programming logic and thereby developing skills in problem solving using Python programminglanguage; To be able to do testing and debugging of code written in Python Emphasize the concepts and constructs rather than on language features.

PRE-REQUISITES

The students are expected to have basic knowledge of programming.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: To learn and understand Python programming basics, looping, control statements andstring manipulations.

CO2: To learn basic concepts of list, tuple and dictionary data structures.

CO3: To learn and know the concepts of Structure and Functions.

CO4: To learn and know the concepts of file handling, exception handling.

CO5: To learn how to analyze and visualize the data.

UNIT I

Introduction to Python: History, Features & Benefits of Python, Structure of a Python Program, Identifiers and Keywords, Concept of Variable, Memory Allocation for Variable, Data Types in Python, Conversion Functions, Operators, Input and Output Function, Control Statements: Conditional Statement, Loops, break, continue and pass statements.

UNIT II

Data Structures & Function: String, Lists, Tuples, Dictionary Data Structure, Built-in Library Function, Method and Operation on these Data Structure. Defining Function, Type of Function Arguments (Required Arguments, Keyword Arguments, Default Arguments, Variable-Length Arguments), Scope of a Variable, Global Vs Local Variable, Python Modules & Packages.

UNIT III

Python Object Oriented Programming: Features of Object Oriented Programming, Creating Classes, Class Variable, Instance Variable, init_() Method, Creating Instance Object, Class Attributes, Access Specifiers in Python, Instance Method Vs Class Method. Inheritance & Polymorphism, Overriding and Overloading Methods.

UNIT IV

Python File Handling, Exception Handling: Opening & Closing Files, File Access Modes, File Object Attributes, Reading and Writing Files, Manipulating File Pointer using seek and tell.

Programming using File Operations. Exception Handling in Python.

UNIT V

Data Analysis & Visualization: Dataset generation, Importing Dataset: Importing and Exporting Data, Basic Insights from Datasets, Cleaning and Preparing the Data: Identify and Handle Missing Values, Basic Visualization Tools, Creating and plotting Maps.

TEXT BOOKS

1. John V Guttag. “Introduction to Computation and Programming Using Python”, Prentice Hall of India.
2. T. Budd, Exploring Python, TMH, 1st Ed, 2011

REFERENCE BOOKS

1. Python Tutorial/Documentation www.python.or 2010
2. Allen Downey, Jeffrey Elkner, Chris Meyers ,How to think like a computer scientist :Learning withPython,Freelyavailableonline.2012
3. http://docs.python.org/3/tutorial/index.html
4. http://interactivepython.org/courselib/static/pythonds

CS-152E	PYTHON PROGRAMMING LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

To learn the fundamentals of Python programming, including syntax, data types, control structures, and functions. Applying Python knowledge to solve real-world problems through hands-on exercises and projects. Enhance problem-solving abilities by writing Python code to address various computational problems.

COURSE OUTCOMES

CO1: Define basic concepts of python programming if statement, loops.

CO2: Define and demonstrate the use of built-in data structures “lists” and “dictionary”.

CO3: Design and implement a program to solve a real-world problem

CO4: Design and implement a program how to handle exceptions and files.

CO5: Design and implement a program for data analysis and visualization.

LIST OF PROGRAMS

Program 1: Programs using if else structure

a) Find the Largest Among Three Numbers.
b) Python Program to Take in the Marks of 5 Subjects and Display the Grade

Program 2: Programs using for and while loop

a) Python Program to check whether given number is Prime Number or not.
b) Python Program to Find the Sum of Digits in a Number.

Program 3: Program using List and String data structure

a) Write Python Program to input a list of integers, (1) display the no of elements in the list (2) display minimum and maximum element in the list (3) display sum of square of all the element in the list (4) (5) add a new element at end and display the list (6) add a new element at given index and display list (7) display the occurrence of given element in the list (8) remove the given element in the list (9) add element from a new list from given list (10) sort the given list & reverse the given list (11) also perform slicing, concatenation and multiplication operation.

b) A fruit seller sells different type of fruits. Type of fruits and corresponding rates are stored in two different lists. Customer can order any type of fruit (one or more type) in any quantity. Iftotal bill of customer is greater than 500, customer is given 10% discount. If any of the fruits required by the customer is not available in the store, then consider the bill amount to be -1. Write a Python program to calculate and display the bill amount.

Program 4: Programs using concept of list, tuple & dictionary

a) A furniture seller sells different type of furniture, Type of Furniture and rates are stored in a dictionary.

Customer can order any type of furniture (one or more type) in any quantity. If total bill of customer is greater than 10,000, customer is given 5% discount. 8% GST is charged on total bill. If any of the furniture required by the customer is not available in thestore, then consider the bill amount to be -1. Write a Python program to calculate and displaythe bill amount.

b) Students name and their corresponding marks are stored in a dictionary. Write a Python program to perform following (1) Display name and marks of each student (2) Display the names of top two scorer (3) display the class average for this course (4) check if the marksfor given student is stored in dictionary or not, if not add the name and marks in the dictionary else display his/her marks (5) delete the name and marks of a given student in the dictionary (6) add name and marks from another dictionary and display combined dictionary.

Program 5: Program using Function in Python:

a) Write Python functions using the concept of Keyword & default arguments and write a program to use them.
b) Write python functions to use the concept of variable length argument & global variable.

Program 6: Program using concept of Class, object, Class Variable, Class Method:

a) Create a class Account with name, account no and balance as attribute and no_of_accountsas class variable. Account no should be generated automatically (starting from 1) using the class variable no_of_account. Add the methods for displaying the account information, depositing given amount, withdrawing given amount and initializer method to initialize the object. Create objects of Account class and call different method to test the class.

b) Create a class Employee with name , empid ,salary as attribute and no_of_ employee and annual_incr (% annual increment) as class variable. empid should be generatedautomatically (starting from 1) using the class variable, no_of_employee. Add the instance methods for displaying the employee information, annually increasing the salary with help ofclass variable annual_incr , class method to change the value of annual_incr and initializer method to initialize the object. Create objects of employee class and call different method to test the class (program using class method).

Program 7: Program using the concept of Inheritance

a) Create a class Polygon to represent a polygon having no of sides and a list having magnitudeof each side as attribute. Add the inputSides() to input sides and displaySides() to display sides as methods. Derive a class Triangle from Polygon and add an additional method displayArea() to display area. Create object of Triangle and call different methods to test the class.

b) Create a class Person having name, age, as attributes, __init__() method to initialize the object and display() to display person information. Derive a class Student from Person having roll no, University name, branch as additional attributes and_init_(), display() to display student information and change_Branch() method. Create object of Student type and call different methods to test the class.

Program 8: Program using the concept of Polymorphism, Operator Overloading

a) In a retail outlet there are two modes of bill Payment (1) Cash : Calculation includes VAT(10%) Total Amount = Purchase amount + VAT (2) Credit card: Calculation includes processing charge and VAT Total

Amount = Purchase amount + VAT (10%) + Processing charge (2%) The act of bill payment is same but the formula used for calculation of total amount differs as per the mode of payment. Can the Payment maker simply call a method and that method dynamically selects the formula for the total amount? Demonstrate this Polymorphic behaviour with code.

b) Write a program to create a class to represent length in feet and inch. Overload the "+" operator to add the two object of length type.

Program 9: Program on file handling and Exception handling in Python

a) Write a python program to write few lines on a file, read it back and create a dictionary having each word in file as keys in dictionary and occurrence of these word as values and print the dictionary.
b) Write a program to show the concept of exceptional handling.

Program 10: Program on Data Analysis and Visualization

Importing and Exporting Data, Basic Insights from Datasets, Cleaning and Preparing theData:

Syllabus

of

3rd & 4th Semester

CS-203E	COMPUTER ARCHITECTURE & ORGANIZATION	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To provide basic knowledge of internals of computer, its architecture, components, terminologies, etc. at minutelevel and ultimately about the working of a digital computer hardware as a whole.

PRE-REQUISITES

Knowledge of data structures, microprocessors and interfacing.

COURSE OUTCOMES

CO1: Understand the theory and architecture of central processing unit. Analyse some of the designissues in terms of speed, technology, cost, performance.

CO2: Design a simple CPU with applying the theory concepts. Use appropriate tools to design verify and testthe CPU architecture.

CO3: Learn the concepts of parallel processing, pipelining and inter-processor communication

CO4: Exemplify in a better way the I/O and memory organization.

CO5: Define different number systems, binary addition and subtraction, 2’s complementrepresentation and operations with this representation.

UNIT I

GENERAL SYSTEM ARCHITECTURE: Functions and block diagram of computer, store program controlconcept, Flynn‘s classification of computers (SISD, MISD, MIMD); CPU, caches, main memory, secondary memory units & I/O; Counters and Designing of counters.

UNIT II

INSTRUCTION SET ARCHITECTURE: Instruction codes, instruction set formats(fixed, variable, hybrid), types of instructions, memory reference, register reference, I/O reference; addressing modes: register, immediate, direct, indirect, indexed; operations in the instruction set; arithmetic and logical, data transfer, control flow; types of interrupts; timing and control; instruction set based classification of processors (RISC, CISC, and their comparison).

UNIT III

BASIC ARITHMETIC AND PROCESSING MODULE: Addition and subtraction of signed

Numbers-Design of fast adders-Multiplication of positive numbers-Signed operand multi-plication and fastmultiplication –integer division

Fundamental Concept-Execution of complete instruction-Multiple bus organization –Hardwired control-Micro Programmed control-Pipelining and its types.

UNIT IV

MEMORY HIERARCHY & SYSTEM: Need for a memory hierarchy (Locality of Reference Principle, memory hierarchy in practice: cache, main memory and secondary memory); Basic Concept Main memory (semiconductor RAM & ROM , Speed, Size and Cost, static & dynamic memory types); cache memory: associative &direct mapped cache organizations.

UNIT V

INTRODUCTION TO PARALLELISM: Goals of parallelism (exploitation of concurrency, throughput enhancement); Amdahl‘s law; instruction level parallelism (pipelining, super scaling-basic features); processorlevel parallelism (multiprocessor systems overview).

TEXT BOOK

1. John P.Hayes, ‘Computer architecture and Organisation’, Tata McGraw-Hill, Third
2. edition, 1998.
3. 2. V.Carl Hama Cher, Zvonko G. Varanesic and Safat G. Zaky, “Computer Organisation“, V edition, McGraw-Hill Inc, 1996.
4. Carpinelli, ―Computer Organization & Architecture‖ Tata McGraw Hill, 2000

CS-205E	**DISCRETE MATHEMATICS and GRAPH THEORY**	**L-T-P**	**Cr**
		3-0-0	**3**

OBJECTIVES

To lay mathematical foundation for the fundamentals of various computational structures such as Boolean algebra, propositional logic, graph and trees.

PRE-REQUISITES: Knowledge of Data Structure

COURSE OUTCOMES

CO1: Perform operations on various discrete structures such as sets, functions, relations, and sequences.

CO2: Ability to solve problems using Counting techniques, Permutation and Combination, Recursion and generating functions

CO3: Ability to solve problems of Recursion and Recurrence Relation

CO4: Understand the various properties of algebraic systems like Rings, Monoids and Groups

CO5: Apply algorithms and use of graphs and trees as tools to visualize and simplify Problems.

UNIT I

SET THEORY: Introduction to set theory; set operations; algebra of sets: duality, finite and infinite sets, classes of sets, power sets, multi sets, Cartesian product, representation of relations, types of relation, equivalence relations and partitions, partial ordering relations and lattices; function and its types, composition of function and relations; cardinality and inverse relations.

UNIT II

PROPOSITIONAL CALCULUS AND TECHNIQUES OF COUNTING: Basic operations: AND (Λ), OR (V), NOT (~), truth value of a compound statement, propositions, tautologies, contradictions, Permutations with and without repetition, combination.

UNIT III

RECURSION AND RECURRENCE RELATION: Polynomials and their evaluation; sequences, introduction to AP, GP and AG series, partial fractions; linear recurrence relation with constant coefficients; homogeneous solutions, particular solutions, total solution of a recurrence relation using generating functions.

UNIT IV

ALGEBRIC STRUCTURES: Definition and examples of a monoid, semigroup, groups and rings; homomorphism, isomorphism and auto morphism; subgroups and normal subgroups; cyclic groups, integral domain and fields; co-sets; Lag range‘s theorem

UNIT V

GRAPHS: Introduction to graphs, directed and undirected graphs; homomorphic and isomorphic graphs; subgraphs; cut points and bridges; multigraph and weighted graph; paths and circuits, shortest path in weighted graphs; Eulerian path and circuits, Hamilton paths and circuits; planar graphs; Euler‘s formula.

TEXT BOOK

Liu C. L., Elements of Discrete Mathematics‖, McGraw Hill, 1989

REFERENCE BOOKS

1. Johnson Bough R., ―Discrete Mathematics‖, 5th Edition, Pearson Education, 2001
2. Graham Ronald, Knuth Donald E. and Patashik Oren, ―Concrete Mathematics: A Foundation for Computer Science‖ , Addison-Wesley, 1989
3. Gersting Judith L., ―Mathematical Structures for Computer Science‖, Computer Science Press, 1993
4. Chtewynd A. and Diggle P.. , Discrete Mathem atics‖ , Modular Mathematics Series, Edward Arnold, London, 1995
5. Lipshutz S ., ―S chaums Outline series: Theory and problems of Probability‖, McGraw Hill Singapore, 1982
6. Kolm an B. and Busby R. C., ―Discrete Mathematical Structures‖, Prentice Hall of India,
7. 1996
8. Trembley and Manohar, ―Discrete Mathematical Structures with Applications to Com puters‖, McGraw Hill, 1995

CS-207E	OPERATING SYSTEM	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To provide the knowledge of internals, different types and purpose of operating systems

PRE-REQUISITES

Knowledge of computer organization and architecture programming skills

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Understand the basics of operating systems like kernel, shell, types and views of operating systems

CO2: Describe the various CPU scheduling algorithms.

CO3: Explain various process synchronization and removing deadlock.

CO4: Use disk management and disk scheduling algorithms for better utilization of external memory.

CO5: Understand various file system.

UNIT I

INTRODUCTION: Introduction to operating system concepts (including multitasking, multiprogramming, multi user, multithreading, etc)., types of operating systems: batch operating system, time-sharing systems, distributed OS, network OS, real time OS, embedded and smart card OS, various operating system services, architecture, system programs and calls.

UNIT II

PROCESS MANAGEMENT: Process concept, Life cycle and implementation of process, Thread usage and implementation in user space and in kernel, process scheduling, operation on processes, CPU scheduling, scheduling criteria, scheduling algorithms -First Come First Serve (FCFS), Shortest-Job-First (SJF), priority scheduling, Round Robin (RR), multilevel feedback queue scheduling.

UNIT III

DEADLOCK: Deadlocks, Deadlock characteristics, prevention, avoidance using banker’s algorithm, detection and recovery;

Process Synchronization: Critical section problems, mutual exclusion with busy waiting, Process synchronization, semaphores: binary and counting semaphores, Classical IPC problems: dining philosophers’ problem, readers-writers problem.

UNIT IV

MEMORY MANAGEMENT: Logical & physical address space, swapping, contiguous memory allocation, non-contiguous memory allocation paging and segmentation techniques, segmentation with paging, virtual memory management - demand paging & page-replacement algorithms, demand segmentation.

UNIT V

I/O AND FILE YSTEMS: I/O hardware, device controllers, interrupt handlers, device drivers, application I/O interface, kernel, transforming I/O requests, performance issues, Different types of files and their access methods, directory structures, various allocation methods, disk scheduling and management and its associated algorithms, introduction to distributed file system.

TEXT BOOK

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 8th Edition, 2014, ISBN10: 0133805913 • ISBN13: 9780133805918
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, John Wiley & Sons ,Inc., 9th Edition,2012, ISBN 9781118063330
3. Maurice J. Bach, “Design of UNIX Operating System”, PHI
4. T1: Silberchatz et al, “Operating System Concepts”, 5th edition, Addison-Wesley, 1998

REFERENCE BOOKS

1. Tom Adelstein and Bill Lubanovic, Linux System Administration, O'Reilly Media, Inc., 1st Edition, 2007.ISBN10: 0596009526 | ISBN13: 9780596009526

CS-209E	DATA STRUCTURE USING C++	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To relay the theoretical and practical fundamental knowledge of most basic data structure like array linked list, stack, queue, tree & graph. To understand the implementation of these data structure to be familiar with basic techniques of algorithm analysis and analysis of the algorithms used for implementation of these data structure.

PRE-REQUISITES: Knowledge of basic computer programming.

COURSE OUTCOMES

CO1: Understand the concept of dynamic memory management, data types, algorithm, Big O notation.
CO2: Understand basic data structures such as arrays, linked list, stack and queue.
CO3: Describe the hash function and concepts of collision and its resolution methods.
CO4: Solve problem involving graph, trees and heaps.
CO5: Apply algorithm for solving problems like sorting, searching, insertion and deletion of data.

UNIT I: INTRODUCTION TO DATA STRUCTURES AND RUNNING TIME:

Definition of data structures and abstract data types; linear vs. non-linear data structure; primitive vs. non-primitive data structure; static and dynamic implementations; arrays, 1,2-dimensional arrays, insertion & deletion in 1-D array; examples and real life applications. Time complexity; Big Oh notation; running times; best case, worst case, average case; factors depends on running time; introduction to recursion.

UNIT II: STACKS AND QUEUES:

Stacks: definition, array based implementation of stacks, examples: infix, postfix, prefix representation; conversions, applications; definition of queues, circular queue; array based implementation of queues.

UNIT III: LINKED LISTS:

Lists; different type of linked Lists; implementation of singly linked list, linked list implementation of stacks and queues; implementation of circular linked list; implementation of doubly linked list, applications.

UNIT IV: TREES AND GRAPHS:

Definition of trees and binary trees; properties of binary trees and implementation; binary traversal pre-order, post-order, in-order traversal; binary search trees: searching, insertion & deletion. Definition of undirected and directed graphs; array based implementation of graphs; adjacency matrix; path matrix implementation; linked list representation of graphs; graph traversal: breadth first traversal, depth first traversal; implementations and applications.

UNIT V: SORTING AND SEARCHING ALGORITHMS:

Introduction, selection, insertions, bubble sort, efficiency of above algorithms; merge sort, merging of sorted arrays and algorithms; quick sort algorithm analysis, heap sort, searching algorithms: straight sequential search, binary search (recursive & non–recursive algorithms).

TEXT BOOK

1. Langsam, Augentem M.J. and Tenenbaum A. M., —Data Structures using C & C++‖, Prentice Hall of India, 2009.
2. R. S.Salariya, Data Structure and Algorithm, Khanna Publications.

REFERENCE BOOKS

1. Aho A. V., Hopcroft J. E. and Ullman T. D., —Data Structures and Algorithms‖, Original Edition, Addison-Wesley, Low Priced Edition, 1983.
2. Horowitz Ellis and S ahni S artaj, —Fundamentals of Data Structures‖, Addison-WesleyPub, 1984.
3. Horowitz, S ahni and Rajasekaran, —Fundamentals of Computer Algorithms‖ 2007.
4. Kruse Robert, —Data Structures and Program Design in C‖, Prentice Hall of India, 1994
5. Lipschetz Jr. Seymour, —Theory & Problems of Data Structures‖, S chaum ‘s Outline,Tata McGraw Hill
6. Weiss Mark Allen, —Data Structures and Algorithms Analysis in C‖, Pearson Education,2000
7. Corm en T . H . et al., —Introduction to Algorithms‖, 2nd Edition, Prentice Hall of India,2001.

CS-257E	**OPERATING SYSTEM LAB**	**L-T-P**	**Cr**
		0-0-2	**1**

OBJECTIVES

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes, threads and their communication.
3. To know the components and management aspects of concurrency management viz. Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.

COURSE OUTCOMES

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

CO1. Learn the basic concepts of operating system, its various types and architecture

CO2. Learn and implement process management issues including process life cycle, scheduling, synchronization and deadlocks

CO3. Learn and implement memory management issues including memory partitioning, memory allocation and virtual memory concept

CO4. Learn and implement files systems and I/O systems including file management, disk management and kernel I/O subsystems.

LIST OF EXPERIMENTS

1. Basics of UNIX commands.
2. Shell programming
3. Implementation of CPU scheduling. a) Round Robin b) SJF c) FCFS d) Priority
4. Implement all file allocation strategies
5. Implement Process System call.
6. Implement I/O system call.
7. Implement Bankers algorithm for Dead Lock Avoidance
8. Implement Producer/Consumer problem using semaphore.
9. Implement the all page replacement algorithms a) FIFO b) LRU c) LFU
10. Implement first fit, best fit algorithm for memory management.
11. To write a program for file manipulation for displays the file and directory in memory

REFERENCE BOOKS

1. Bach Maurich, “Design of the Unix Operating System”, Prentice Hall of India, 1986
2. Prato Stephen, “Advanced Unix Programmer’s Guide”, BPB Publications, 2006
3. Das Sumitabha, “Unix- Concept and Applications”, Tata McGraw Hill, 2002.

CS-259E	DATA STRUCTURE USING C++ LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

Familiarize students with various fundamental data structures like arrays, linked lists, stacks, queues, trees, graphs, and hash tables. Teach students how to use appropriate data structures to solve computational problems efficiently.

COURSE OUTCOMES

CO1: Students will be able to understand and implement basic and advanced data structures (such as stacks, queues, trees, graphs).

CO2: Gain the ability to implement algorithms like searching (linear, binary), sorting (quick, merge, bubble), and traversal (DFS, BFS) efficiently

CO3: Enhance the problem-solving skills by choosing the appropriate data structure for a given problem.

CO4: Hands-on experience in coding data structures in C++ improves practical coding skills and familiarity with the language.

LIST OF EXPERIMENTS

ARRAY OPERATIONS

1. Write a program to insert an element at given position in linear array
2. Write a program to insert an element in sorted array.
3. Write a program to delete an element from given position in linear array
4. Perform following operations on matrices using functions only
   a) Addition b) Subtraction c) Multiplication d) Transpose

SEARCHING

5. Search an element in a linear array using linear search.
6. Using iteration and recursion concepts write programs for finding the element in thearray using Binary Search Method

RECURSION

7. Write a program to compute factorial of given number using recursion
8. Write as program to solve Tower of Hanoi problem using recursion
9. Write a program to find power of given number using recursion

STACK & QUEUE

10. Write a program for static implementation of stack
11. Write a program for dynamic implementation of queue
12. Write a program for static implementation of circular queue
13. Write a program for dynamic implementation of queue

14. Write a program to evaluate a postfix operation

LINKED LIST

15. Create a linear linked list & perform operations such as insert, delete at end , atbeg & reverse the link list
16. Create a circular linked list & perform search, insertion & delete operation
17. Create a doubly linked list & perform search, insertion & delete operation

SORTING ALGORITHMS

18. Write program to implement Bubble, Insertion & selection sort.
19. Write program to implement quick sort
20. Write program to implement merge sort

TEXT BOOK

1. A.K. Sharma – Data structure Using C, 2nd edition pearson 2013
2. Langsam, Augentem M.J. and Tenenbaum A. M., ―Data Structures using C & C++‖,Prentice Hall of India, 2009.

REFERENCE BOOKS

1. R. S. Salaria -Data Structure Using C
2. Kruse Robert, ―Data Structures and Program Design in C‖, Prentice Hall of India,1994
3. Lipschitz Jr. Seymour, ―Theory & Problems of Data Structures‖, Schaum‘s Outline, 2ndEdition, Tata McGraw Hill

CS-202E	DATABASE MANAGEMENT SYSTEM	L-T-P	Cr
		3-0-0	3

OBJECTIVES

To understand the different issues involved in the design and implementation of a database system.

PRE-REQUISITES Knowledge of Operating system

COURSE OUTCOMES

CO1: For a given query write relational algebra expressions for that query and optimize the developed expressions

CO2: For a given specification of the requirement design the databases using ER method and normalization

CO3: For a given specification construct the SQL queries for Open source and Commercial DBMS -MYSQL, ORACLE, and DB2

CO4: For a given query optimize its execution using Query optimization algorithms

UNIT 1

Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations

UNIT-2

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server. Relational database design: Domain and data dependency, Armstrong's axiom, Normal forms, Dependency preservation, Lossless design. Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

UNIT-3

Storage strategies: Indices, B-trees, hashing.

UNIT-4

Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery

UNIT-5

Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

TEXT BOOK

"Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

REFERENCE BOOKS

1. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
2. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education

3 .Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

CS-204E	DESIGN AND ANALYSIS OF ALGORITHM	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To relay the theoretical and practical aspects of design of algorithms

PRE-REQUISITES

Knowledge of fundamentals of basic computer programming for implementing algorithms

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Define the basic concepts of algorithms and analyze the performance of algorithms.

CO2: Discuss various algorithm design techniques for developing algorithms.

CO3: Discuss various searching, sorting and graph traversal algorithms.

CO4: Understand NP completeness and identify different NP complete problems.

CO5: Discuss various advanced topics on algorithms.

UNIT I

BRIEF REVIEW: Growth of functions, Asymptotic Notations, Representation of Graphs, Breadth First Search, Depth First Search and Data Structures for Disjoint Sets.

UNIT II

DIVIDE AND CONQUER: General method; binary search; merge sort; quick sort; Strassen‘s matrix multiplication algorithms and analysis of algorithms for these problems.

UNIT III

GREEDY METHOD: General method; knapsack problem, job sequencing with deadlines; minimum spanning trees Algorithm of Kruskal’s and Prim’s; single source paths and analysis of these problems.

UNIT IV

DYNAMIC PROGRAMMING AND BACK TRACKING: General method; optimal binary search trees; O/I knapsack; the traveling salesperson problem, 8 queens ‘problem; graph coloring; Hamiltonian cycles

UNIT V

NP HARD AND NP COMPLETE PROBLEMS: Basic concepts; Cook‘s theorem; NP hardgraph and NP scheduling problems; some simplified NP hard problems.

TEXT BOOK

Horowitz Ellis and Sahni Sartaj, ―Fundamental of Computer Algorithms‖, Galgotia Publications, 1978

REFERENCE BOOKS

1. Cormen Thomas H., Leiserson Charles E. and Rivest Ronald L., ―Introduction to Algorithms‖, Tata McGraw Hill, 1990
2. Aho A. V. and Hopcroft J. E., ―The Design and Analysis of Computer Algorithm‖, AddisonWesley, 1974
3. Berlion P., and Bizard P., Algorithms – The Construction, Proof and Analysis of Programs,John Wiley & Sons, 1986.
4. Bentley J. L., ―Writing Efficient Programs‖, Prentice Hall of India, June 1982.
5. Goodman S. E. and Hedetnieni, ―Introduction to Design and Analysis of Algorithm‖, McGraw Hill, 1997
6. Trembley Jean Paul and Bunt Richard B., ―Introduction to Computers Science - AnAlgorithms Approach‖, Tata McGraw Hill, 2002
7. Knuth Donald E., ―Fundamentals of Algorithms: The Art of Computer Programming‖, Vol.l, Naresh Publications, 1985
8. Goodrich Michael T. and Roberto Tamassia, ―Algorithm Design: Foundations, Analysis &Internet Examples‖, Wiley Student Ed., 2002

CS-206E	OBJECT ORIENTED PROGRAMMING USING JAVA	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To relay the theoretical and practical aspects of java

PRE-REQUISITES

Formal introduction to Java programming language

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Define the basic concepts of algorithms and analyze the performance of algorithms.

CO2: Discuss various algorithm design techniques for developing algorithms.

CO3: Discuss various searching, sorting and graph traversal algorithms.

CO4: Understand NP completeness and identify different NP complete problems.

CO5: Discuss various advanced topics on algorithms.

UNIT I

Introduction to Java : Basics of Java programming, Data types, Variables, Operators, Control structures
S including selection, Looping, Java methods, Overloading, Math class, Arrays in java.

UNIT II

Objects and Classes : Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, StringBuffer, File, this reference.

UNIT III

Inheritance and Polymorphism : Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.

UNIT IV

Event and GUI programming : Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing.

UNIT V

I/O programming and Multithreading in java: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try-catch-finally, Collections in java, Introduction to JavaBeans and Network Programming..

TEXT BOOK

Horowitz Ellis and Sahni Sartaj, ―Fundamental of Computer Algorithms‖, Galgotia Publications, 1978

REFERENCE BOOKS

- Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson.
- Programming in Java, Sachin Malhotra & Saurabh Chaudhary, Oxford University Press.
- Murach's Beginning Java 2, Doug Lowe, Joel Murach and Andrea Steelman, SPD.
- Core Java Volume-I Fundamentals, Eight Edition, Horstmann & Cornell, Pearson Education.
- The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, TMH.
- Java Programming, D. S. Malik, Cengage Learning

CS-208E	ARTIFICIAL INTELLIGENCE	L-T-P	Cr
		3-0-0	3

OBJECTIVES

To introduce about artificial intelligence approaches to problem solving, various issues involved and application areas

PRE-REQUISITES: Knowledge of neural networks, data structures

COURSE OUTCOMES

CO1: Demonstrate fundamental understanding of artificial intelligence (AI) and expert systems. Solve basic AI based problems

CO2: Define the concept of Artificial Intelligence and Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.

CO3: Apply AI techniques to real-world problems to develop intelligent systems.

CO4: Select appropriately from a range of techniques when implementing intelligent systems.

CO5: Discuss the basics of ANN and different optimizations techniques

UNIT I

INTRODUCTION TO AI AND SEARCH TECHNIQUES: Foundation and history of AI; data, information and knowledge; AI problems and techniques – AI programming languages, problem space representation with examples; blind search strategies, breadth first search, depth first search, heuristic search techniques: hill climbing: best first search, A * algorithm AO* algorithm, Minimax search procedure for Game Playing.

UNIT II

KNOWLEDGE REPRESENTATION ISSUES AND TECHNIQUES: Predicate logic; representing knowledge using rules. Semantic nets, partitioned nets, parallel implementation of semantic nets; frames, forward and backward chaining; frame based systems.

UNIT III

REASONING UNDER UNCERTAINITY: Reasoning under uncertainty, non monotonic reasoning; Review of probability; Baye's probabilistic interferences and Dumpster Shafer theory; statistical reasoning, fuzzy reasoning.

UNIT IV

PLANNING & LEARNING: Goal stack planning; non linear planning, hierarchical planning representation for planning; partial order planning algorithm. Basic concepts of Learning ; rote learning, learning by taking advices, learning by problem solving, learning from examples, discovery as learning, learning by analogy; explanation based learning; neural nets; genetic algorithms.

UNIT V

EXPERT SYSTEM AND APPLICATIONS OF ARTIFICIAL INTELLIGENCE: Expert systems: rule based systems architecture: Principles of natural language processing: knowledge acquisition concepts; AI application to robotics, and current trends in intelligent systems; parallel and distributed AI.

TEXT BOOK

Rich Elaine and Knight Kevin, ―Artificial Intelligence 3rd Edition, Tata McGraw Hill, 1991

REFERENCE BOOKS

1. Nilson Nils J., ―Artificial Intelligence‖, McGraw-Hill, New York 1971
2. Russell Stuart and Norvig Peter, ―Artificial Intelligence: A Modern Approach‖, Prentice Hall of India, 1998
3. Negnevitsky, ―Artificial Intelligence: A Guide to Intelligent System ‖, Pearson Education, 2004.
4. Patterson O. W., ―Introduction to Artificial Intelligence & Expert Systems‖, Prentice Hall of India, 1996.
5. Winston Patrick Henry, ―Artificial Intelligence, 3rd Edition, Addition Wesley, 1992
6. Clockson & Mellish, ―Programming PROLOG‖, 3rd Edition, Narosa Publications, 2002

CS-210E	COMPUTER NETWORK	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To develop an understanding of modern network architectures from a design and performance perspective.

PRE-REQUISITES

The major cocepts involve in wide area networks(WANs),local area networks (LANs) and wireless LANs

COURSE OUTCOMES

CO1: Explain the functions of the different layer of the OSI Protocol.

CO2: Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.

CO3: For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) design it based on the market available component

CO4: For a given problem related TCP/IP protocol developed the network programming.

CO5: Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools

UNIT-1:DATA COMMUNICATION COMPONENTS

Representation of data and its flow Networks , Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum

UNIT-2: DATA LINK LAYER AND MEDIUM ACCESS SUB LAYER

Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CD,CDMA/CA

UNIT-3: NETWORK LAYER

Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP andDHCP– Delivery, Forwarding and Unicast Routing protocols.

UNIT-4: TRANSPORT LAYER

Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm

UNIT-5: APPLICATION LAYER

Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography

TEXT BOOKS

Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGrawHill.

REFERENCES:

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGrawHill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
3. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
4. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
5. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America

CS-212E	THEORY OF COMPUTATION	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To relay the theoretical and practical aspects of formal languages

PRE-REQUISITES

Formal languages and automata theory deals with the concepts of automata, formal languages, grammar, computability and decidability

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: To use basic concepts of formal languages of finite automata techniques.

CO2: To Design Finite Automata's for different Regular Expressions and Languages.

CO3: To Construct context free grammar for various languages

CO4: To solve various problems of applying normal form techniques, push down automata and Turing Machines

CO5: To participate in GATE, PGECET and other competitive examinations.

UNIT I

FINITE AUTOMATA (FA): Introduction, Deterministic Finite Automata (DFA) -Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion. .

UNIT II

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions. REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, Closure properties of regular languages

UNIT III

CONTEXT FREE GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL (Proof's omitted).

UNIT IV

PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA. TURING MACHINES (TM): Formal definition and behaviour, Languages of a TM, TM as accepters and TM as a computer of integer functions, Types of TMs.

UNIT V

RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL): Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP..

TEXT BOOK

John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages andComputation, 3rdedition, Pearson Education, India.

REFERENCE BOOKS

K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India,

CS-216E	BASIC & ADVANCED EXCEL	L-T-P	Cr
		0-0-4	2

OBJECTIVE

Learn how to structure and organize data efficiently using rows, columns, and worksheets. Understand how to create basic charts (e.g., pie charts, bar charts, line graphs) to visually represent data.

COURSE OUTCOMES

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

CO1: Create and manage Excel workbooks and worksheets, including importing and organizing data effectively.

CO2: Navigate within workbooks using advanced search and navigation tools, and customize the workbook environment to suit personal preferences.

CO3: Apply custom data formats, data validation, and advanced conditional formatting to manage and present data clearly and accurately.

CO4: Create and manage tables, utilize sorting and filtering options, and enhance data analysis by removing duplicates and summarizing data.

CO5: Design, format, and analyze data using charts and objects, applying various styles, layouts, and modifications to improve data visualization.

List of Experiments

1. Creating and Organizing Workbooks with Imported Data
2. Customizing Worksheet Tabs, Themes, and Layouts
3. Navigating and Searching Data within Workbooks
4. Applying Data Validation Rules for Controlled Input
5. Implementing Conditional Formatting to Highlight Data
6. Creating and Managing Excel Tables
7. Sorting and Filtering Data in Excel Tables
8. Using Basic Formulas and Functions for Data Analysis
9. Applying Advanced Functions for Enhanced Data Analysis
10. Creating and Customizing Charts for Data

CS-252E	DATABASE MANAGEMENT SYSTEM LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

To provide knowledge about implementation of practical aspects of database i.e. creation of tables andapplying queries using SQL queries

COURSE OUTCOMES

The students undergoing this course will be able:

CO1: To know the basics of structured query language

CO2: To aware about applying different queries on database structured in the form of tables

CO3: To learn about the different SQL queries performed using operators as well asconstraints

CO4: To create views from created table to further organize the data

CO5: To learn about the basic operations of relational algebra

LIST OF EXPERIMENTS/EXERCISES

1. Introduction to SQL.
2. Write a query for:
   (i) Creation of table.
   (ii) Insertion of data into table
   (iii) Displaying the data of table.
   (iv) Deletion of data from table
   (v) Updating the data
   (vi) Modifying the structure of a table.

3. Finding unique names of all salesmen, deletion of the structure of a table, use of delete command with conditions, updating records of a table with conditions, altering structure of a table and changing sizeof existing column in the table
4. Arithmetic operators, logical operators and pattern matching operator.
5. Key constraints: primary key constraints, foreign key constraints, not null constraints and uniqueconstraints; use of check constraints.
6. Aggregate and mathematical functions: count, count(*), Avg, max, min, sum, lower, upper, power ,sqrt.
7. Creating views from single and multiple tables, drop views and creating index on the table and dropthem.
8. Binary operations in Relational Algebra: Union, Intersection, Set Difference, Join, Cartesian product.
9. Grouping of data into tables and listing records in ascending order or descending order.
10. Creation of sequences and explain use of sequences.
11. Access permissions in SQL.

CS-256E	OBJECT ORIENTED PROGRAMMING USING JAVA LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

OOP allows breaking down large problems into smaller, manageable pieces using classes and objects, promoting modularity in program design.

COURSE OUTCOMES

CO1: Code reusability and modularity speed up development by allowing developers to use existing components.

CO2: OOP mirrors real-world entities and their interactions, improving problem-solving by creating more intuitive designs.

CO3: The abstraction and encapsulation mechanisms make Java code more readable and easier to understand.

CO4: OOP practices encourage clean and organized code, reducing errors and improving quality.

LIST OF EXPERIMENTS

- Program to define a structure of a basic JAVA program
  Program to define the data types, variable, operators, arrays and control structures.
- Program to define class and constructors. Demonstrate constructors..
- Program to define class, methods and objects. Demonstrate method overloading
- Program to define inheritance and show method overriding.
- Program to demonstrate Packages.
- Program to demonstrate Exception Handling
- Program to demonstrate Multithreading
- Program to demonstrate I/O operations
- Program to demonstrate Network Programming.
- Program to demonstrate Applet structure and event handling
- Program to demonstrate Layout managers

TEXT-BOOKS

Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson.

REFERENCE-BOOKS

- Programming in Java, Sachin Malhotra & Saurabh Chaudhary, Oxford University Press
- . Murach’s Beginning Java 2, Doug Lowe, Joel Murach and Andrea Steelman, SPD
- . Core Java Volume-I Fundamentals, Eight Edition, Horstmann & Cornell, Pearson Education.
- The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, TMH.
- Java Programming, D. S. Malik, Cengage Learning.

CS-258E	ARTIFICIAL INTELLIGENCE LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

To push the boundaries of AI by exploring new algorithms, models, and techniques. This can include developing new machine learning algorithms, improving neural network architectures, or creating novel methods for natural language processing.

COURSE OUTCOMES

CO1 Explain artificial intelligence, its characteristics and its application areas.

CO2 Formulate real-world problems as state space problems, optimization problems or constraint satisfaction problems

CO3 Select and apply appropriate algorithms and AI techniques to solve complex problems.

CO4 Design and develop an expert system by using appropriate tools and techniques.

LIST OF EXPERIMENTS

1. Study of Python programming language.
2. Write a program to find out route distance between two cities using Python.
3. Write a program to implement Tower of Hanoi using Python.
4. Write a program to calculate factorial of a number using Python.
5. Write a program to print the list of customer having different colored cars with price and model available using Python.
6. Write a program to implement water jug problem using Python.
7. Write a program to implement Breadth First Search using Python
8. Write a program to implement Depth First Search using Python
9. Write a program to solve 8-Queens problem using Python.
10. Write a program to solve Monkey Banana problem using Python.

Syllabus

of

5th & 6th Semester

CS-301E	MACHINE LEARNING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

The main objective of this course is to enabling the student with basic knowledge on the techniques to build an intellectual machine for making decisions behalf of humans. This course covers the techniques on how to make learning by a model, how it can be evaluated, what are all different algorithms to construct a learning model.

PRE-REQUISITES: Knowledge of python programming and probability and statistics.

COURSE OUTCOMES

CO1: Understand the structure, syntax, and semantics of the python language.

CO2: Appreciate the importance of visualization in the data analytics solution.

CO3: Identify various machine learning algorithms and terminologies and perform data pre-processing using standard ML library.

CO4: Understand appropriate unsupervised learning algorithms for performing clustering and dimensionality reduction.

CO5: Implement probabilistic graphical models for suitable applications.

UNIT I: CONCEPTS OF PYTHON PROGRAMMING:

Python data structures, Control statements, Functions, Object Oriented programming concepts using classes, objects and methods, Exception handling, Implementation of user- defined Modules and Package, File handling in python.

UNIT II: INTRODUCTION TO MACHINE LEARNING:

Machine Learning Fundamentals –Types of Machine Learning -Supervised, Unsupervised, Reinforcement- The Machine Learning process.
Terminologies in ML- Testing ML algorithms: Overfitting, Training, Testing and Validation Sets Confusion matrix -Accuracy metrics- ROC Curve- Basic Statistics: Averages, Variance and Covariance, The Gaussian- The Bias-Variance trade off- Applications of Machine Learning.

UNIT III: SUPERVISED LEARNING:

Regression: Linear Regression – Multivariate Regression- Classification: Linear Discriminant Analysis,
Logistic Regression- K-Nearest Neighbor classifier.
Decision Tree based methods for classification and Regression- Ensemble methods.

UNIT IV: UNSUPERVISED LEARNING:

Clustering- K-Means clustering, Hierarchical clustering - The Curse of Dimensionality - Dimensionality Reduction - Principal Component Analysis - Probabilistic PCA- Independent Components analysis.

UNIT V: PROBABILISTIC GRAPHICAL MODELS:

Bayesian Networks - Learning Naive Bayes classifiers-Markov Models – Hidden Markov Models. Sampling – Basic sampling methods – Monte Carlo -Reinforcement Learning.

TEXT-BOOKS

1. Machine Learning: An Algorithmic Perspective by Stephen Marsland, Chapman and Hall/CRC.
2. T. Mitchell, Machine Learning, McGraw Hill.
3. M. Gopal, Applied Machine Learning, McGraw Hill.
4. Sutton R. S. and Barto, A. G., Reinforcement Learning: An Introduction, The MIT Press (2017).

REFERENCE-BOOKS

1. Introduction to Machine Learning by Ethem Alpaydin, PHI Learning.
2. M. Evangelia, Supervised and Unsupervised Pattern Recognition, CRC Press.
3. G. James, D. Witten, T. Hastie, R. Tibshirani, Introduction to Statistical Learning, Springer

CS-303E	COMPILER DESIGN	L-T-P	Cr
		3-1-0	4

OBJECTIVE

To design the front end of the compiler, scanner, parser, intermediate code generator, object code generator, and the parallel compilation strategies.

PRE-REQUISITES

Formal Grammars and Languages theory deals with the concepts of compilation techniques.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Use compiler construction tools and describes the Functionality of each stage of compilation process

CO2: Construct Grammars for Natural Languages and find the Syntactical Errors/Semantic errors during the compilations using parsing techniques

CO3: Analyze different representations of intermediate code.

CO4: Construct new compiler for new languages.

CO5: Participate in GATE, PGECET and other competitive examinations

UNIT – I

FRONT END OF COMPILERS The structure of Compiler – Lexical analysis: Role of Lexical analyzer, Specification and recognition of tokens, Syntax Analysis: Top down parsing, Bottom up parsing, LR Parsers: SLR, CLR, and LALR.

UNIT – II

INTERMEDIATE CODE GENERATION Syntax Directed Definitions, Evaluation orders for syntax directed definitions, Syntax Directed Translation schemes, Intermediate languages : Three address code, Syntax tree, Postfix code – Declarations – Type checking – Expression translation – Back patching

UNIT – III

OBJECT CODE GENERATION Storage organization, Stack allocation space, Access to non-local data on the stack, Heap management - Issues in code generation - Design of code generator - Register allocation and assignment – Instruction selection by tree rewriting – Optimal code generation for expressions – Dynamic programming code generation.

UNIT – IV

CODE OPTIMIZATION Basic blocks and Flow graphs – Optimization of basic blocks – Principal sources of optimizations – Data flow analysis – Constant propagation – Partial redundancy elimination - Peephole optimizations.

UNIT – V

PARALLELIZING COMPILER Basic concepts and examples – Iteration spaces – Affine array indexes – Data reuse – Array data dependence - Finding synchronization free parallelism – Synchronization between parallel loops, Locality optimizations.

TEXT BOOKS:

Alfred V. Aho, Monica S.Lam, Ravi Sethi, Jeffrey D.Ullman, “Compilers : Principles, Techniques and Tools”, Second Edition, Pearson Education, 2008.

REFERENCES:

1. Randy Allen, Ken Kennedy, “Optimizing Compilers for Modern Architectures: A Dependence-based Approach”, Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, “Advanced Compiler Design and Implementation”, Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper and Linda Torczon, “Engineering a Compiler”, Morgan Kaufmann Publishers Elsevier Science, 2004.
4. V. Raghavan, “Principles of Compiler Design”, Tata McGrawHill Education Publishers, 2010. 5. Allen I. Holub, “Compiler Design in C”, Prentice-Hall software series, 1993

CS-305E	AGILE DEVELOPMENT & METHODOLOGY	L-T-P	Cr
		3-0-0	3

Objectives

Upon completing the course, students will be able to:
Familiarize themselves with the fundamental concepts and principles of Agile Software Development.
Understand and apply Agile estimation, planning techniques, and requirement management.

Course Outcomes

After successfully completing this course, the students will be able to:

CO1: Define and explain the common characteristics and principles of Agile development processes.

CO2: Compare and contrast various Agile methodologies, such as Scrum, Extreme Programming (XP), and Lean Software Development.

CO3: Differentiate between Agile software development models and traditional, plan-driven process models.

CO4: Evaluate and identify which software project characteristics are most suitable for Agile processes.

UNIT-I: History of Agile Methodologies and Core Concepts

- Overview of traditional software development (Waterfall) and its limitations.
- Evolution of Agile and why it was needed.
- Differences between Agile and Lean development methodologies.
- Core principles of Lean and Agile.
- Definition and principles of XP.
- Practices of XP (Pair programming, Test-driven development, continuous integration).

UNIT-II: Agile Estimation, Planning, and Requirements Management

- Techniques for Agile estimation (Story Points, Planning Poker, T-Shirt sizing).
- How to plan Agile projects (Release Planning, Sprint Planning).
- The shift from traditional requirement gathering to Agile requirements.
- Writing effective User Stories.
- INVEST criteria for good User Stories.

UNIT-III: Tracking Agile Projects and Risk Management

- Time tracking in Agile (Burndown charts, Burnup charts).
- Techniques for tracking progress and velocity of Agile teams.
- Principles of Lean in software development.
- Differences between Lean and traditional development models.
- Overview of model-driven development and its role in Agile.

UNIT-IV: Agile Tools and Enterprise Agility

- Introduction to Agile tools (Jira, Trello, Asana).

- How to use tools for managing backlogs, tracking sprints, and collaboration.
- Concepts of enterprise agility and how large organizations can adopt Agile principles.
- DSDM principles, phases, and how it fits within Agile frameworks.

Textbooks:

1. Agile Development with Scrum by Ken Schwaber & Mike Beedle
2. Integrating Agile Development in the Real World by Peter Schuh

Reference Books:

1. Agile Software Development – The Cooperative Game by Alistair Cockburn
2. Succeeding With Agile, Software Development Using Scrum by Mike Cohn

CS-307E	WEB TECHNOLOGY & SERVICES	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To understand the concepts and architecture of the World Wide Web.

To understand and practice Markup Language.

To understand and practice Embedded Dynamic Scripting on Client-side Internet Programming.

To understand and practice Web Development Techniques on client-side.

PRE-REQUISITES

Basics of programming, Structure and HTML Tags, Images, List, Tables, Anchors and FormElements

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Explain the history of the internet and related internet concepts that are vital in understanding web development.

CO2: Discuss the insights of internet programming and implement complete application over theweb.

CO3: Demonstrate the important HTML tags for designing static pages and separate design fromcontent using Cascading Style sheet.

CO4: Utilize the concepts of JavaScript and Java

CO5: Use web application development software tools i.e. Ajax, PHP and XML etc. and identifythe environments currently available on the market to design web sites.

UNIT I

INTRODUCTION TO WWW: Introduction to Computer networks - Internet Standards – Introduction to WWW – WWW Architecture – SMTP – POP3 – File Transfer Protocol - Overview of HTTP, HTTP request – response –– Generation of dynamic web pages.

UNIT II

UI DESIGN: HTML5: What is HTML5 - Features of HTML5 – Semantic Tags – New Input Elements and tags - Media tags (audio and video tags) – Designing Graphics using Canvas API - Drag and Drop features – Geolocation API - Web storage (Session and local storage).

CSS3: What is CSS3 – Features of CSS3 – Implementation of border radius, box shadow, imageborder, custom web font, backgrounds - Advanced text effects(shadow) - 2D and 3D Transformations - Transitions to elements - Animations to text and elements

UNIT III

RESPONSIVE WEB DESIGN (RWD): Responsive Design: What is RWD – Introduction to RWD Techniques – Fluid Layout, Fluid Images and Media queries - Introduction to RWD Framework .

TWITTER BOOTSTRAP – Bootstrap Background and Features - Getting Started with Bootstrap - Demystifying Grids – Off Canvas - Bootstrap Components - JS Plugins - Customization

UNIT IV

INTRODUCTION TO JAVASCRIPT: Introduction - Core features - Data types and Variables

- Operators, Expressions and Statements - Functions & Scope - Objects - Array, Date and Math related Objects - Document Object Model - Event Handling – Browser Object Model - Windows and Documents - Form handling and validations.

OBJECT-ORIENTED TECHNIQUES IN JAVASCRIPT - Classes – Constructors and Prototyping (Sub classes and Super classes) – JSON – Introduction to AJAX.

UNIT V

INTRODUCTION TO JQUERY: Introduction – jQuery Selectors – jQuery HTML - Animations – Effects – Event Handling – DOM – jQuery DOM Traversing, DOM Manipulation

– jQuery AJAX

TEXT BOOKS

1. Harvey & Paul Deitel & Associates, Harvey Deitel and Abbey Deitel, "Internet and World Wide Web - How To Program", Fifth Edition, Pearson Education, 2011.
2. Achyut S Godbole and Atul Kahate, "Web Technologies", Second Edition, Tata McGraw Hill, 2012.

REFERENCE BOOK

1. Thomas A Powell, Fritz Schneider, "JavaScript: The Complete Reference", Third Edition, Tata McGraw Hill, 2013.
2. David Flanagan, "JavaScript: The Definitive Guide, Sixth Edition", O'Reilly Media, 2011
3. Bear Bibeault and Yehuda Katz, "jQuery in Action", January 2008
4. Web link for Responsive Web Design - https://bradfrost.github.io/this-is-responsive/
5. Ebook link for JavaScript - https://github.com/jasonzhuang/tech_books/tree/master/js

CS-351E	MACHINE LEARNING LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

To develop new machine learning algorithms or improve existing ones to enhance performance, scalability, and efficiency. To train and evaluate machine learning models on various datasets to understand their strengths and weaknesses.

COURSE OUTCOMES

CO1: Innovative algorithms that offer better accuracy, reduced computation time, or improved generalization.

CO2: Well-evaluated models with documented performance metrics, including accuracy, precision, recall, and F1 score.

CO3: Functional ML-driven solutions that address real-world challenges and demonstrate practical utility.

LIST OF EXPERIMENTS

1. Installation of Python / Python Libraries.
2. Data pre-processing using Python Machine Learning libraries.
3. Design a model to predict the housing price using Multivariate Linear Regression.
4. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
5. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions.
6. Build a classifier using Logistic Regression, k- Nearest Neighbor to classify whether the given user will purchase a product or not from a social networking dataset.
7. Segment a customer dataset based on the buying behavior of customers using K-means.
8. Implement the decision tree using publically available dataset.
9. Dimensionality reduction of any CSV/image dataset using Principal Component Analysis.
10. Build an email spam classifier using SVM.

TEXT-BOOKS

1. Machine Learning: An Algorithmic Perspective by Stephen Marsland, Chapman and Hall/CRC.
2. T. Mitchell, Machine Learning, McGraw Hill.
3. M. Gopal, Applied Machine Learning, McGraw Hill.
4. Sutton R. S. and Barto, A. G., Reinforcement Learning: An Introduction, The MIT Press (2017).

REFERENCE-BOOKS

1. Introduction to Machine Learning by Ethem Alpaydin, PHI Learning.
2. M. Evangelia, Supervised and Unsupervised Pattern Recognition, CRC Press.
3. G. James, D. Witten, T. Hastie, R. Tibshirani, Introduction to Statistical Learning, Springer

CS-357E	WEB TECHNOLOGY & SERVICES LAB	L-T-P	Cr
		0-0-2	1

OBJECTIVE

To develop and refine web applications using modern technologies and frameworks. This includes front-end, back-end, and full-stack development.

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Explain the history of the internet and related internet concepts that are vital in understanding web development.

CO2: Discuss the insights of internet programming and implement complete application over theweb.

CO3: Demonstrate the important HTML tags for designing static pages and separate design fromcontent using Cascading Style sheet.

CO4: Utilize the concepts of JavaScript and Java

CO5: Use web application development software tools i.e. Ajax, PHP and XML etc. and identifythe environments currently available on the market to design web sites.

LIST OF EXPERIMENTS

1 Design a web page using Physical and Logical tags of HTML.
2 Design a web page using
2.1 Ordered List2.2
2.2 Unordered List
2.3 Nested Lists

3 Design a web page to show the use of image as a hyperlink
4 Design a web-page using frames and linking
5 Design a class Time Table using tables in HTML.
6 Code to create a bookmark.
7 Design a web-page showing the use of forms using HTML 4.01 and HTML 5 Tags.
8 Design a page using basic tags of HTML 5.0.
9 Design a web-page using style sheets (External, Internal and Inline)
10 Write a Program to print if the no is even or odd using JavaScript
11 Input a number and find the difference of the sum of factors and non-factors.
12 WAP in JavaScript to print the pattern12345

1234

123

12

1

13 WAP to Accept an Array of 10 numbers and display the sum of elements.
14 WAP to find greatest of all elements of an array
15 Design a web-page to show different validation checking using Java Script
16 WAP in PHP code for calculating S.I
17 WAP to Calculate factorial of a number
18 WAP to print the table of 10.
19 WAP to print the sum of diagonal elements
20 WAP to enter 5 elements each from Array1 and Array2 and print these elements using thirdarray.
21 WAP to show database connectivity using PHP and Mysql.

CS-302E	SOFTWARE ENGINEERING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To provide basic knowledge of properties of software and its development processes, software quality, CASE tools, etc.

PRE-REQUISITES

Knowledge of computer programming, principles of management

COURSE OUTCOMES

The students undergoing this course will be able to:

CO1: Plan a software engineering process life cycle, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements

CO2: Able to elicit, analyze and specify software requirements through a productive working relationship with various stakeholders of the project

CO3: Analyze and translate a specification into a design, and then realize that design practically, using an appropriate software engineering methodology.

CO4: Know how to develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice

CO5: Able to use modern engineering tools necessary for software project management, time management and software reuse.

UNIT I

INTRODUCTION: Definition and Emergence of Software Engineering, Evolving Role of Software, Software Life Cycle Models, Software Characteristics, Applications, Software Product, Software Process, Software Crisis, Software Myths.

UNIT II

SOFTWARE PROJECT MANAGEMENT: Project management concepts, software process and project metrics project planning, project size estimation metrics, project estimation techniques, empirical estimation techniques, COCOMO- a heuristic estimation techniques, staffing level estimation, team structures, staffing, risk analysis and management, project scheduling and tracking.

UNIT III

REQUIREMENTS ANALYSIS AND SPECIFICATION: Requirements engineering, system modeling and simulation, analysis principles: modeling, partitioning, software, prototyping: methods and tools; specification principles, representation, the software requirements specification and reviews analysis modeling: data modeling, functional modeling and informationflow: data flow diagrams, behavioral modeling; the mechanics of structured analysis: creating entity/ relationship diagram, data flow model, control flow model, the control and process specification.

UNIT IV

SYSTEM DESIGN AND COMPUTERAIDEDSOFTWARE ENGINEERING: Design Process: design and software quality, design principles; design concepts: abstraction, refinement, modularity, software architecture, control hierarchy, structural partitioning, software procedure, information hiding; functional independence, cohesion, coupling; design heuristics for effective modularity; design model; design documentation, architectural design: software architecture, CASE, building blocks; integrated case environments and architecture, repository.

UNIT V

TESTING AND MAINTENANCE: Software testing techniques, software testing fundamentals: objectives, principles, testability; test case design, white box testing, basis path testing: control structure testing: black box testing, testing for specialized environments, architectures and applications. software testing strategies: verification and validation, unit testing, integration testing, validation testing, alpha and beta testing; system testing, acceptance testing debugging approaches; software re-engineering, reverse engineering, restructuring, forward engineering, Software maintenance, Adaptive , corrective and perfective, software reliability: measures of reliability and availability, software safety.

TEXT BOOK

Pressman Roger S., ―Software Engineering – A Practitioner‘s Approach‖ , McGraw Hill, 2004

REFERENCE BOOKS

1. Jalote P ankaj, ―An Integrated Approach to Software Engineering‖, 3rd edition, Narosa Book Distributors Private Ltd, 2005
2. Mall Ra jib, ―Fundamentals ofSoftware Eng ineering ‖, Prentice Hall of India,2003
3. Sommerville Ian, ―Software Engineering‖, 8th edition, Addison Wesley, 2007
4. Gustafson David, ―Software Eng ineering ‖, Tata McGraw Hill, 2002
5. Behforooz Ali and Hudson Frederick J., ―Software Engineering Fundamentals‖, Oxford University press, John Wiley & Sons, 2005.

CS-304E	BIG DATA ANALYTICS	L-T-P	Cr
		3-0-0	3

OBJECTIVES

To provide an overview of an emerging field of big data analytics. To make students familiar with the tools required to manage and analyze big data like Hadoop, No SQL, Map-Reduce. To teach the fundamental techniques and principles in achieving analytics with scalability and streaming capability on both structured and unstructured data.

COURSE OUTCOMES

CO1: Understand the key issues in big data management and its associated applications for business decisions and strategy.

CO2: Develop problem solving and critical thinking skills in fundamental enabling techniques like Hadoop, MapReduce and NoSQL in big data analytics.

CO3: Collect, manage, store, query and analyze various forms of Big Data.

CO4: Interpret business models and scientific computing paradigms and apply software tools for big data analytics.

CO5: Adapt adequate perspectives of big data analytics in various applications like recommender systems, social media applications etc.

MODULE I: INTRODUCTION TO BIG DATA Types of Digital Data, Characteristics of Data, Evolution of Big Data, Definition of Big Data, Challenges with Big Data, 3Vs of Big Data, NonDefinitional traits of Big Data - Business Intelligence vs. Big Data, Coexistence. Big Data Analytics: Classification of analytics, Terminologies in Big Data. Big Data Processing Architectures, Data Warehouse, ReEngineering the Data Warehouse, Big data learning approaches.

MODULE II: NTRODUCTION TO HADOOP Features of Hadoop, Advantages, Versions, Physical Architecture of Hadoop, Analyzing Data with Hadoop, Overview of Hadoop Eco systems, Hadoop distributions, Hadoop vs. SQL, RDBMS vs. Hadoop, Components of Hadoop, Concept of HDFS, Hadoop Streaming

MODULE III: HDFS (HADOOP DISTRIBUTED FILE SYSTEM) AND MAP REDUCE Distributed File Systems: Physical Organization of Compute Nodes, Large-Scale FileSystem Organization. The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives MapReduce: The Map Tasks, Grouping by Key, The Reduce Tasks, Combiners, Details of MapReduce Execution, Coping with Node Failures. Algorithms Using MapReduce: MatrixVector Multiplication by MapReduce, RelationalAlgebra Operations, Computing Selections by MapReduce

MODULE IV: NO SQL DATABASE Introduction to NoSQL, NoSQL Business Drivers, NoSQL Data Architecture Patterns: Keyvalue stores, Graph stores, Column family (Bigtable)stores, Document stores, Variations of NoSQL architectural patterns, NoSQL Case Study, NoSQL solution for big data, Understanding the types of big data problems; Analyzing big data with a shared-nothing architecture; Choosing distribution models: master-slave versus peer-to-peer; NoSQL systems to handle big data problems.

MODULE V: HADOOP ECO SYSTEM Pig: Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators. Hive: Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL, Tables, Querying Data and User Defined Functions. Hbase:HBasics

Text Books:

1. Anand Rajaraman and Jeff Ullman "Mining of Massive Datasets", Cambridge University Press,

2. Alex Holmes "Hadoop in Practice", Manning Press, Dreamtech Press.

3. Dan McCreary and Ann Kelly "Making Sense of NoSQL" – A guide for managers and the rest of us, Manning Press.

References:

1. Chuck Lam, "Hadoop in Action", Dreamtech Press

2. Judith Hurwitz, Alan Nugent, Dr. Fern Halper, Marcia Kaufman, "Big Data for Dummies", Wiley India 2. Michael Minelli, Michele Chambers, Ambiga Dhiraj, "Big Data Big Analytics: Emerging Business Intelligence And Analytic Trends For Today's Businesses", Wiley India

3. Paul Zikopoulos, Chris Eaton, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data', McGraw Hill Education.

CS-352E	**SOFTWARE ENGINEERING LAB**	**L-T-P**	**Cr**
		0-0-2	**1**

OBJECTIVE

To provide practical exposure to the various phases of SDLC, such as requirements analysis, design, coding, testing, and maintenance. To understand and implement core software engineering principles (modularity, abstraction, encapsulation, etc.) in real-world scenarios.

COURSE OUTCOME

CO1: Students will be able to understand and implement each phase of the software development life cycle, from requirements gathering to maintenance.

CO2: Become proficient in using version control systems (e.g., Git), IDEs, and tools for software design, testing, and documentation.

CO3: Experience in working collaboratively within teams to develop software projects, manage tasks, and solve conflicts effectively.

CO4: Learn how to manage a project by setting timelines, allocating tasks, and ensuring deliverables are met, while understanding basic project management methodologies.

LIST OF EXPERIMENTS

1. Phases in software development project, overview, need,
2. To assign the requirement engineering tasks
3. To perform the system analysis : Requirement analysis, SRS
4. To perform the function oriented diagram : DFD and
5. To perform the user's view Analysis: Use case diagram
6. To draw the structural view diagram: Class diagram, object
7. To draw the behavioural view diagram: Sequence diagram,
8. To draw the behavioural view diagram: State-chart diagram,
9. To draw the implementation view diagram: Component
10. To draw the environmental view diagram : Deployment
11. a)Library Management System

    b)Automated banking system

    c)Airline reservation system

d)Employee management application

CS-358E	MONGO DB	L-T-P	Cr
		0-0-4	2

OBJECTIVE

To introduce students to NoSQL databases and their significance as an alternative to traditional relational databases. To learn about MongoDB as a document-oriented NoSQL database. To explore MongoDB's architecture, data storage model, and BSON (Binary JSON) format.

COURSE OUTCOMES

Upon successful completion of the course, the student will be able to:

CO1: Installing and configuring MongoDB in Windows

CO2: Perform all database operations using MongoDB

CO3: Develop applications by integrating MongoDB with Java/PHP.

LIST OF EXPERIMENTS

1. MongoDB installation and configuration in Windows.

2. Demonstrate how to create and drop a database in MongoDB.

3. Creating the Collection in MongoDB on the fly.

4. Creating collection with options before inserting the documents and drop the collection created.

5. MongoDB insert document
   a. Insert single document
   b. Insert multiple documents in collection

6. Querying all the documents in JSON format and Querying based on the criteria.

7. MongoDB update document
   a. Using update() method.
   b. Using save() method.

8. MongoDB delete document from a collection.
   a. Using remove() method.
   b. Remove only one document matching your criteria
   c. Remove all documents

9. MongoDB Projection

10. limit(), skip(), sort() methods in MongoDB

11. MongoDB indexing
    - a. Create index in MongoDB
    - b. Finding the indexes in a collection
    - c. Drop indexes in a collection
    - d. Drop all the indexes

12. MongoDB with Java and PHP
    - a. Create a simple application that uses MongoDB with Java
    - b. Create a simple application that uses MongoDB with PHP

Syllabus

of

7th & 8th Semester

CS-401E	DEEP LEARNING	L-T-P	Cr
		3-0-0	3

OBJECTIVES

The objective of this course is to cover the fundamental of neural networks as well as some advanced topics such as recurrent neural networks, long short term memory cells and convolutional neural networks. The course also requires students to implement programming assignments related to these topics.

COURSE OUTCOMES

CO1: Understand the fundamentals and current usage of the TensorFlow library for deep learning research and the graphical computational model of TensorFlow

CO2: Understand the context of neural networks and deep learning

CO3: Design recurrent neural networks with attention mechanisms for natural language classification, generation, and translation.

CO4: Perform regularization, training optimization, and hyperparameter selection on deep models.

CO5: Explore the parameters for neural networks

UNIT 1

BASICS: Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Linear separability. Convergence theorem for Perceptron Learning Algorithm.

FEEDFORWARD NETWORKS: Multilayer Perceptron, Gradient Descent, Backpropagation, Empirical Risk Minimization, regularization, autoencoders.

UNIT II

DEEP NEURAL NETWORKS: Difficulty of training deep neural networks, Greedy layerwise training.
BETTER TRAINING OF NEURAL NETWORKS: Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).

UNIT III

RECURRENT NEURAL NETWORKS: Back propagation through time, Long Short Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs
CONVOLUTIONAL NEURAL NETWORKS: LeNet, AlexNet.

UNIT IV

GENERATIVE MODELS: Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.

UNIT V

RECENT TRENDS: Variational Autoencoders, Generative Adversarial Networks, Multi-task Deep Learning, Multi-view Deep Learning
APPLICATIONS: Vision, NLP, Speech (just an overview of different applications in 2-3 lectures)

TEXT BOOKS

Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016.

REFERENCES

1. Neural Networks: A Systematic Introduction, Raúl Rojas, 1996
2. Pattern Recognition and Machine Learning, Christopher Bishop, 2007

CS-403E	R PROGRAMMING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To understand the fundamentals of R as a programming language. Learn basic syntax, operators, data types (vectors, lists, data frames), and control structures (loops, conditionals). Explore built-in functions for data manipulation.

COURSE OUTCOMES

CO1: Access online resources for R and import new function packages into the R workspace

CO2: Import, review, manipulate and summarize data-sets in R

CO3: Explore data-sets to create testable hypotheses and identify appropriate statistical tests

CO4: Perform appropriate statistical tests using R

CO5: Create and edit visualizations with R

UNIT – I

INTRODUCTION: Overview of R, R data types and objects, reading and writing data, sub setting R Objects, Essentials of the R Language, Installing R, Running R, Packages in R, Calculations, Complex numbers in R, Rounding, Arithmetic, Modulo and integer quotients, Variable names and assignment, Operators, Integers, Factors, Logical operations

UNIT – II

Control structures, functions, scoping rules, dates and times, Introduction to Functions, preview of Some Important R Data Structures, Vectors, Character Strings, Matrices, Lists, Data Frames, Classes

VECTORS: Generating sequences, Vectors and subscripts, Extracting elements of a vector using subscripts, Working with logical subscripts, Scalars, Vectors, Arrays, and Matrices, Adding and Deleting Vector Elements, Obtaining the Length of a Vector, Matrices and Arrays as Vectors Vector Arithmetic and Logical Operations, Vector Indexing, Common Vector Operation

UNIT – III

LISTS: Creating Lists, General List Operations, List Indexing Adding and Deleting List Elements, Getting the Size of a List, Extended Example: Text Concordance Accessing List Components and Values Applying Functions to Lists, DATA FRAMES, Creating Data Frames, Accessing Data Frames, Other Matrix-Like Operations

UNIT - IV

FACTORS AND TABLES, Factors and Levels, Common Functions Used with Factors, Working with Tables, Matrix/Array-Like Operations on Tables , Extracting a Subtable, Finding the Largest Cells in a Table, Math Functions, Calculating a Probability, Cumulative Sums and Products, Minima and Maxima, Calculus, Functions for Statistical Distributions

UNIT - V

OBJECT-ORIENTED PROGRAMMING: S Classes, S Generic Functions, Writing S Classes, Using

Inheritance, S Classes, Writing S Classes, Implementing a Generic Function on an S Class, visualization, Simulation, code profiling, Statistical Analysis with R, data manipulation

TEXT BOOKS

1. R Programming for Data Science by Roger D.Peng
   The Art of R Programming by Prashanthsingh, VivekMourya, CengageLearning India.

CS-403E	INTERNET OF THINGS	L-T-P	Cr
		3-0-0	3

OBJECTIVE

It enables the students to describe what IoT is and how it works today, Recognise the factors that contributed to the emergence of IoT and Design and program IoT devices.

PRE-REQUISITES:

Basic knowledge of passive electrical & electronics components, basic programming such as C language, accessibility to the Hardware such as Arduino UNO & Arduino IDE.

COURSE OUTCOMES

The students undergoing this course will be able:

CO1: To Use real IoT protocols for communication

CO2: To Secure the elements of an IoT device

CO3: To Design an IoT device to work with a Cloud Computing infrastructure

CO4: Transfer IoT data to the cloud and in between cloud providers

CO5: To Define the infrastructure for supporting IoT deployments

UNIT I

IOT: What is the IoT and why is it important? Elements of an IoT ecosystem, Technology drivers, Business drivers, Trends and implications, Overview of Governance, Privacy and Security Issues

UNIT II

IOT PROTOCOLS: Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFIDProtocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4–BACNet Protocol– Modbus – KNX – Zigbee– Network layer – APS layer – Security

UNIT III

IOT ARCHITECTURE:

IoT Open source architecture (OIC)- OIC Architecture & Design principles- IoT Devices and deployment models- IoTivity : An Open source IoT stack - Overview- IoTivity stack architecture- Resource model and Abstraction.

UNIT IV

WEB OF THINGS:

Web of Things versus Internet of Things – Two Pillars of the Web – Architecture Standardization for WoT– Platform Middleware for WoT – Unified Multitier WoT Architecture – WoT Portals and Business Intelligence.

UNIT V

IOT APPLICATIONS:

IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, SmartApplications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc

Textbook:

1. Honbo Zhou, "The Internet of Things in the Cloud: A Middleware Perspective", CRCPress, 2012.
2. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), "Architecting the Internetof Things", Springer, 2011.
3. David Easley and Jon Kleinberg, "Networks, Crowds, and Markets: Reasoning About aHighly Connected World", Cambridge University Press, 2010.
4. Olivier Hersent, David Boswarthick, Omar Elloumi , "The Internet of Things – Keyapplications and Protocols", Wiley, 2012.

References Books:

1. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)",1st Edition, VPT, 2014
2. Francis da Costa, "Rethinking the Internet of Things: A Scalable Approach to ConnectingEverything", 1st Edition, Apress Publications, 2013 Cuno Pfister, Getting Started with the Internet of Things, O"Reilly Media, 2011, ISBN: 978-1- 4493-9357-1

CS-405E	**PREDICTIVE ANALYTICS MODULAR**	**L-T-P**	**Cr**
		3-0-0	**3**

OBJECTIVE

To teach the students and make them Understand Basics of Predictive Analytics, Understand Data types and Variable types, Understand Basic Modeling and work on Missing Data

PRE-REQUISITES:

Should have the awareness of Associate Analytics & Big data concepts Able to understand the Tools & technology related to the maintenance of Data bases.

COURSE OUTCOMES

The students undergoing this course will be able:

CO1: Understand how Predictive analytics can be used in the IT environment.

CO2: Students will grasp the meaning, benefits of Predictive analytics

CO3: Students will understand analyze prediction business capabilities using Time series/Forecasting methods and Extract features

UNIT I

Introduction to Predictive Analytics & Linear Regression :- What and Why Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of data and variables, Data Modeling Techniques, Missing imputations etc. Need for Business Modeling, Regression — Concepts, Blue property-assumptions-Least Square Estimation, Variable Rationalization, and Model Building etc

UNIT II

Logistic Regression (Model Theory, Model fit Statistics, Model Conclusion, Analytics applications to various Business Domains etc. Regression Vs Segmentation — Supervised and Unsupervised Learning, Tree Building — Regression, Classification, Over fitting, Pruning and complexity, Multiple Decision Trees etc.

UNIT III

Objective Segmentation Regression Vs Segmentation — Supervised and Unsupervised Learning, Tree Building — Regression, Classification, Over fitting, Pruning and complexity, Multiple Decision Trees etc. Develop Knowledge, Skill and Competences (NOS 9005) Introduction to Knowledge skills & competences, Training & Development, Learning & Development, Policies and Record keeping. etc.

UNIT IV

Time Series Methods I Forecasting, Feature Extraction: Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height. Average, Energy etc and Analyze for prediction.

UNIT V

Working with Documents : Standard Operating Procedures for documentation and knowledge sharing, Defining purpose and scope documents, Understanding structure of documents — case studies, art ides, white papers, technical reports, minutes of meeting etc., Style and format, Intellectual Property and

Copyright, Document preparation tools — Vision, PowerPoint, Word, Excel etc., Version Control, Accessing and updating corporate knowledge base, Peer review and feedback. Books and References
: **Text Books**

- Student’s Handbook for Associate Analytics-Ill.

Reference Books

- Gareth James’ Daniela Witten Trevor Hastie Robert Tibshirani. An Introduction to Statistical Learning with Applications in R

CS-403E	R PROGRAMMING	L-T-P	Cr
		3-0-0	3

OBJECTIVE

To understand the fundamentals of R as a programming language. Learn basic syntax, operators, data types (vectors, lists, data frames), and control structures (loops, conditionals). Explore built-in functions for data manipulation.

COURSE OUTCOMES

CO1: Access online resources for R and import new function packages into the R workspace

CO2: Import, review, manipulate and summarize data-sets in R

CO3: Explore data-sets to create testable hypotheses and identify appropriate statistical tests

CO4: Perform appropriate statistical tests using R

CO5: Create and edit visualizations with R

UNIT – I

INTRODUCTION: Overview of R, R data types and objects, reading and writing data, sub setting R Objects, Essentials of the R Language, Installing R, Running R, Packages in R, Calculations, Complex numbers in R, Rounding, Arithmetic, Modulo and integer quotients, Variable names and assignment, Operators, Integers, Factors, Logical operations

UNIT – II

Control structures, functions, scoping rules, dates and times, Introduction to Functions, preview of Some Important R Data Structures, Vectors, Character Strings, Matrices, Lists, Data Frames, Classes
VECTORS: Generating sequences, Vectors and subscripts, Extracting elements of a vector using subscripts, Working with logical subscripts, Scalars, Vectors, Arrays, and Matrices, Adding and Deleting Vector Elements, Obtaining the Length of a Vector, Matrices and Arrays as Vectors Vector Arithmetic and Logical Operations, Vector Indexing, Common Vector Operation

UNIT – III

LISTS: Creating Lists, General List Operations, List Indexing Adding and Deleting List Elements, Getting the Size of a List, Extended Example: Text Concordance Accessing List Components and Values Applying Functions to Lists, DATA FRAMES, Creating Data Frames, Accessing Data Frames, Other Matrix-Like Operations

UNIT - IV

FACTORS AND TABLES, Factors and Levels, Common Functions Used with Factors, Working with Tables, Matrix/Array-Like Operations on Tables , Extracting a Subtable, Finding the Largest Cells in a Table, Math Functions, Calculating a Probability, Cumulative Sums and Products, Minima and Maxima, Calculus, Functions for Statistical Distributions

UNIT - V

OBJECT-ORIENTED PROGRAMMING: S Classes, S Generic Functions, Writing S Classes, Using Inheritance, S Classes, Writing S Classes, Implementing a Generic Function on an S Class, visualization, Simulation, code profiling, Statistical Analysis with R, data manipulation

TEXT BOOKS

2. R Programming for Data Science by Roger D.Peng
3. The Art of R Programming by Prashanthsingh, VivekMourya, CengageLearning India.

CS-451E	**DEEP LEARNING LAB**	**L T P**	**Cr**
		0-0-2	**1**

OBJECTIVE

Understand deep learning architectures (CNNs, RNNs, GANs). Implement neural networks using TensorFlow, Keras, and PyTorch. Train, validate, and optimize deep learning models. Handle large datasets, preprocessing, and augmentation. Deploy deep learning models in production.

COURSE OUTCOMES

CO1: Ability to design and implement deep learning models.

CO2: Proficiency in tools like TensorFlow, Keras, and PyTorch.

CO3: Skill in model evaluation and hyperparameter tuning.

CO4: Experience developing small-scale deep learning projects.

CO5: Improved problem-solving and critical thinking skills.

PRACTICAL EXERCISES:

1. Implement Simple Programs like vector addition in TensorFlow.
2. Implement a simple problem like regression model in Keras.
3. Implement a perceptron in TensorFlow/Keras Environment.
4. Implement a Feed-Forward Network in TensorFlow/Keras.
5. Implement an Image Classifier using CNN in TensorFlow/Keras.
6. Implement a Transfer Learning concept in Image Classification.
7. Implement an Autoencoder in TensorFlow/Keras.
8. Implement a SimpleLSTM using TensorFlow/Keras.
9. Implement an Opinion Mining in Recurrent Neural network.
10. Implement an Object Detection using CNN.
11. **11.** Mini Project

CS-453E	**R PROGRAMMING LAB**	**L-T-P**	**Cr**
		0-0-2	**1**

OBJECTIVE

Help students become proficient in using R for statistical analysis and data manipulation. Familiarize students with the R programming environment, including RStudio. Practice importing, cleaning, and transforming data from various sources.

COURSE OUTCOMES

CO1: Write efficient R code for data analysis tasks.

CO2: Develop custom functions and use control structures like loops and conditionals to manipulate data.

CO3: Import, clean, and manipulate datasets from various sources (CSV, Excel, databases, etc.).

CO4: Have a foundation for exploring advanced topics like machine learning, time series analysis, and more, using R.

LIST OF EXPERIMENTS

1. Write an R-Program to print HelloWorld
2. Write an R-Program to take input fromuser.
3. Write an R-Program to demonstrate working with operators (Arithmetic, Relational, Logical, Assignmentoperators).
4. Write an R Program to Check if a Number is Odd orEven
5. Write an R Program to check if the given Number is a PrimeNumber
6. Write an R Program to Find the Factorial of aNumber
7. Write an R Program to Find the Factors of aNumber
8. Write an R Program to Find the Fibonacci sequence Using RecursiveFunction
9. Write an R Program to Make a SimpleCalculator
10. Write an R Program to Find L.C.M of twonumbers
11. Write an R Program to create a Vector and to access elements in a Vector
12. Write an R Program to create a Matrix and access rows and columns usingfunctions
13. *colnames()* and *rownames()* .
14. Write an R Program to create a Matrix using *cbind()* and *rbind()*functions.
15. Write an R Program to create a Matrix from a Vector using ***dim()***function.
16. Write an R Program to create a List and modify itscomponents.
17. Write an R Program to create a DataFrame.
18. Write an R Program to access a Data Frame like aList.
19. Write an R Program to access a Data Frame like aMatrix.
20. Write an R Program to create aFactor.
21. Write an R Program to Access and Modify Components of aFactor.

22. Write an R Program to create an S3 Class and S3Objects.
23. Write an R Program to write a own generic function in S3Class.
24. Write an R Program to create an S4 Class and S4Objects.
25. Write an R Program to write a own generic function in S4 Class.
26. Write an R Program to create Reference Class and modify its Methods.