<u>**Four Year Undergraduate Programme**</u>

<u>**Subject: Computer Science**</u>

Programme name	Eligibility Criteria of the programme, if any	Semester	Course name	Course code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)	Internal marks	External Marks	Practical
						L	T	P				
FYUGP in Computer Science (Major/ Minor)	H.S (Science) with Mathematics as a subject, securing at least 33% marks	1	Introduction to C-Programming **Major 1 &** **Minor1** (**for Minor stream) &** **Minor 1 (*For Major in other subject*)**	COM0101004	4	3	0	1	No	30	45	25
		2	Computer Organization **Major 2 &** **Minor2 (for Minor stream &** **Minor 2 (*For Major in other subject*)**	COM0200104	4	4	0	0	No	40	60	0

		3	Object Oriented Programming using C++ **Major 3 & Minor3 (for Minor stream) Minor 3 (*For Major in other subject*)**	COM0300104	4	3	0	1	No	30	45	25
			Data Structure Using C/C++ **Major 4 & Minor4(For Minor stream)**	COM0300204	4	3	0	1	No	30	45	25
		4	Database Management System **Major 5 & Minor 5 & Minor 4(*For Major in other subject*)**	COM0400204	4	3	0	1	No	30	45	25
			Mathematical Foundation of Computer Science **Major6**	COM0400304	4	4	0	0	No	40	60	0
			Operating System **(Major7) & Minor 6**	COM0400404	4	3	0	1	No	30	45	25
			Java Programming OR Python Programming	COM0400504 COM0400604	4	3	0	1	No	30	45	25

			Major 8									
		5	Computer Networks **Major 9**	COM0500104	4	3	0	1	No	30	45	25
			Software Engineering **Major 10 &** **Minor 7 (for Minor stream)** & **Minor 5** (***For Major in other subject***)	COM0500204	4	4	0	0	No	40	60	0
			Web Technologies **Major 11 &** **Minor 8 (for Minor stream)**	COM0500304	4	4	0	0	No	30	45	25
		6	Automata Theory and Languages **Major 12 &** **Minor 9 (for Minor stream)**	COM0600104	4	4	0	0	No	40	60	0
			Cloud Computing **OR** Compiler Design **Major13**	COM0600204 COM0600304	4	4	0	0	No	40	60	0
			Artificial Intelligence **OR** Computer Graphics	COM0600404	4	3	0	1	No	30	45	25

			OR Data Mining and Warehousing **Major14 & Minor 10 (for Minor stream)** **Minor 10 (for Minor stream) & Minor 6** (***For Major in other subject***)	COM0600504 COM0600604								
			Project and Presentation **Major15 &** **Minor 11(for Minor stream)**	COM0600704	4	4	0	0	No	40	60	0

Programme name (AEC/VAC/MDC/SEC)	Eligibility Criteria of the programme, if any	Semester	Course name	Course code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)	Internal marks	External Marks
						L	T	P			
SEC	No	1	Computer Fundamentals & Application Sotware		3	2	0	1		20	30
SEC	No	2	HTML Programming		3	1	0	2		20	30

Students need to take 5 papers in 7th and 8th semester each

Template for Computer Science (Fourth Year) (FYUGP in Computer Science with Honours)

Programme name	Eligibility Criteria of the programme, if any	Semester	Course name	Course code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)	Internal marks	External Marks	Practical
						L	T	P				
FYUGP in Computer Science (Honors)	Computer Science as a MAJOR or MINOR Subject upto 3rd Year		Research Methodology (C)	COM0700104	4	4	0	0	No	40	60	0
			Fundamentals of Machine Learning (E)	COM0700204	4	3	0	1	No	30	45	25
			Advance Operating System (E)	COM0700304	4	3	0	1	No	30	45	25
	H.S (Science) with Mathematics as a		Advanced Computer Organization and Architecture	COM0700404	4	4	0	0	No	40	60	

	subject securing 33% marks	7 Any four Electives (E)	(E)									
			Cryptography and Network Security (E)	COM0700504	4	4	0	0	No	40	60	
			Advanced DBMS (E)	COM0700604	4	3	0	1	No	30	45	25
		8 Any four Electives (E)	Advanced Data Structure (E)	COM0800104	4	3	0	1	No	30	45	25
			Embedded Systems (E)	COM0800204	4	4	0	0	No	40	60	0
			Mobile Application Development (E)	COM0800304	4	3	0	1	No	30	45	25
			System Administration and Networking (E)	COM0800404	4	3	0	1	No	30	45	25
			Mobile Computing (E)	COM0800504	4	4	0	0	No	40	60	0

			Project and Presentation (C)	COM0800604	4	0	0	4	No	40	60	0

Template for Computer Science (Fourth Year) (FYUGP in Computer Science Honours with Research)

Programme name	Eligibility Criteria of the programme, if any	Semester	Course name	Course code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)	Internal marks	External Marks	Practical
						L	T	P				
FYUGP in Computer Science (Honors with Research)	Computer Science as a MAJOR or MINOR Subject upto 3rd Year		Research Methodology (C)	COM0700104	4	4	0	0	No	40	60	0
			Fundamentals of Machine Learning (E)	COM0700204	4	3	0	1	No	30	45	25
			Advance Operating System (E)	COM0700304	4	3	0	1	No	30	45	25
			Advanced Computer Organizatio	COM0700404	4	4	0	0	No	40	60	

		7 Any four Electives (E)	n and Architecture (E)									
			Cryptography and Network Security (E)	COM0700504	4	4	0	0	No	40	60	
			Advanced DBMS (E)	COM0700604	4	3	0	1	No	30	45	25
		8	Dissertation (c)	COM-Dissertation	16				No			
			Project and Presentation	COM0800604	4	0	0	4	No	40	60	0

1-year PG Programme: The programme is consists of three (3) curricular components which are mutually exclusive; (A) Research Components (B) Course work Components (C) Research and Course work components. A student can avail ONLY one component.

A. Research:

Semester	Course level	Course work (CORE*)	Research component/Project	Total Credits

9th	500	--	Dissertation phase 1	20
10th	500	--	Dissertation phase 2	20

OR

B. Course work:

Semester	Course level	Course work (CORE*)	Research component/Project	Total Credits
9th	500	5 courses with 4 credits each (5 X4=20)	--	20
10th	500	5 courses with 4 credits each (5 X4=20)	--	20

OR

C. Research and Course work:

Semester	Course level	Course work (CORE*)	Research component/Project	Total Credits
9th	500	5 courses with 4	--	20

		credits each (5 X4=20)		
10th	500	--	Dissertation of 20 credits	20

One Year Postgraduate Programme

Subject: Computer Science

Template for Computer Sciencce One Year Postgraduate Programme (Course Work)

Programme name	Eligibility Criteria of the programme, if any	Semester	Course name	Course code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)	Internal marks	External Marks	Practical	Remarks
						L	T	P					
1-year PG Programme in Computer Science	FYUGP Computer Science (Honors or Honors with Research		Artificial Neural Networks (E)	COM0900104	4	3	0	1	No	30	45	25	Any 3
			Soft Computing (E)	COM0900204	4	4	0	0	No	40	60	0	
			Block chain Technology (E)	COM0900304	4	4	0	0	No	40	60	0	

	FYUGP in Computer Science/ Information Technology/ Computer Application/ Computer Science Engineering	9	Digital Image Processing (E)	COM0900404	4				No	40	60	0	
			Advance Machine Learning Techniques (E)	COM0900504	4	3	0	1	No	30	45	25	Any 2
			Big Data Analytics	COM09000604	4	3	0	1	No	30	45	25	
			Bioinformatics	COM0900704	4	4	0	0	No	40	60	0	
		10	Optimization Techniques	COM1000104	4	4	0	0	No	40	60	0	Any 3
			Data Science	COM1000204	4	4	0	0	No	40	60	0	
			Deep Learning	COM1000304	4	3	0	1	No	30	45	25	
			Distributed Systems	COM1000404	4	4	0	0	No	40	60	0	
			Speech Processing	COM1000504	4	3	0	1	No	30	45	25	
			Geographical Information System	COM1000604	4	4	0	0	No	40	60	0	Any 2
			Natural Language Processing	COM1000704	4	4	0	0	No	30	45	25	

			Remote Sensing	COM1000804	4	3	0	1	No	30	45	25	
			Knowledge Representation and Reasoning	COM1000904	4	3	0	1	No	30	45	25	

Subject: Information Technology

Template for Information Technology One Year Postgraduate Programme (Course Work)

Programme name	Eligibility Criteria of the programme, if any	Semester	Course name	Course code	Credits	Credit distribution of the course			Pre-requisite of the course (if any)	Internal marks	External Marks	Pratical	
						L	T	P					
1-year PG Programme in Information Technology	FYUGP Computer Science (Honors or Honors with Research	9	Artificial Neural Networks	INF0900104	4	3	0	1	No	30	45	25	Any 3
			Soft Computing	INF0900204	4	4	0	0	No	40	60	0	
	FYUGP in Computer Science/ Information Technology/ Computer Application/ Computer Science Engineering		Block chain Technology	INF0900304	4	4	0	0	No	40	60	0	
			Digital Image Processing	INF0900404	4	3	0	1	No	30	45	25	
			Advanced Web Development Technique	INF0900504	4	3	0	1	No	30	45	25	Any 2
			Multimedia and Graphic design	INF0900604	4	2	0	2	No	30	45	25	

			Ethics in Information Technology	INF0900704	4	4	0	0	No	40	60	0	
		10	Optimization Techniques	INF1000104	4	4	0	0	No	40	60	0	Any 3
			Data Science	INF1000204	4	4	0	0	No	40	60	0	
			Deep Learning	INF1000304	4	3	0	1	No	40	60	25	
			Distributed Systems	INF1000404	4	4	0	0	No	40	60	0	
			Speech Processing	INF1000504	4	3	0	1	No	30	45	25	
			Video Editing and Animation	INF1000604	4	2	0	2	No	40	60	0	Any 2
			Cyber Security	INF1000704	4	4	0	0	No	30	45	25	
			Internet of Things	INF1000804	4	4	0	0	No	30	45	25	
			Software Testing	INF1000904	4	4	0	0	No	40	60	0	

			Augmented Reality	INF101004	4	4	0	0	No	40	60	0	

COM0101004: Introduction to C-Programming

1. **Learning Outcomes**: At the end of the course, students will be able to:
   (a) Understand the basics of C programming like data types and operators
   (b) Understand and write program in C to implement conditions, loops, functions
   (c) Work on arrays, strings and basic file operations

2. **Prerequisites**: NIL

3. **Semester**: 1

4. **Course type**: Compulsory

5. **Course level**: 100-199

6. **Theory credit**: 3

7. **Practical credit**: 1

8. **Number of required hours**:

   a) Theory: 45 hrs (45 classes)

   b) Practical: 30 hrs (15 classes)

   c) Non Contact: NIL

9. **Reference books**:

   (a) B.S. Gottfried, "Schaum's Outline of Theory and Problems of Programming with C", Mcgraw-Hill, 2007.
   (b) B. Kernighan, D. Ritchie, "The C Programming Language", Second Edition, Prentice Hall, 1988
   (c) E. Balaguruswami, "Programming in ANSI C", 2nd Ed., Tata McGraw Hill, 2004.
   (d) P. Greg, D. Miller. "C Programming: Absolute Beginner's Guide", 3rd ed. Que, 2016.

10. **Detailed Syllabus:**

A. Theory

Unit 1: Getting started with C programming (10 Lectures)

Introduction to programming languages- High-level vs low level languages, compiled vs interpreted languages. Structure of a C program. Introduction to Header files. Main function and a simple program execution. Compiling and executing a program. C tokens – keywords, identifiers, constants, operators. Statements and expressions in C. Basic data types in C - integers, floats, doubles, characters. Void. Size and range of values of data types. Variables. Constants – integer constant, real constant, character constant, string constant. Declaration and initialization of variables and constants. Assigning values to variables. Operators in C – binary and unary operators. Arithmetic, assignment, logical, comparison, bitwise and conditional operators. Order of precedence of operators. Associativity of operators. Input and output statements – getchar(), getc(), getch(), putchar(), putc(), puts(), scanf(), printf(), format specifiers. Typecasting.

Unit 2: Control Structures in C (9 Lectures)

Control Structures in C. Basic programming constructs- Sequence, selection and iteration. Conditional statements – if, else, switch case. Nested conditions. Loops – for loop, while loop, do- while loop. Using loop for counting iterations. Using while loop for indefinite iterations. Nested loops. Break and continue statements.

Unit 3: Arrays and Strings (8 Lectures)

Introduction to Arrays. Declaration and initialization of arrays. Accessing array elements. Multidimensional arrays. Introduction to Strings. Declaration and initialization of strings. String input and output in C.

Unit 4: Functions and Pointers (9 Lectures)

Introduction to Pointers. Pointer declaration and initialization. Pointers and addresses. Pointers and Arrays. Basic concept of dynamic memory allocation, malloc(), calloc(). Introduction to functions. Function declaration and definition. Return types of function. Function arguments. Function calling – call by value vs call by reference. Passing an array as argument to a function. Basic concept of recursion.

Unit 5: Introduction to Structures and Unions (4 Lectures)

Basic concept of Structures and Unions in C. Structure declaration and initialization. Union declaration and initialization. Difference between structures and unions.

Unit 6: File Processing in C (5 Lectures)

Basic concept of file handling. Opening and closing file using fopen() and fclose(). Binary vs text files. Reading and writing files – fgets(), fscanf(), fprintf(). Random access to files.

B. List of Practical

(This is a suggestive list only. Questions need not be restricted to this list. The practical are advised to be performed in Linux environment)

1. Write a program in C to print "Hello World"
2. Write a program to take input of two numbers and print their sum, product, difference.
3. Write a program to find the smallest or greatest of three numbers given as input.
4. Write a program to print the sum and product of digits of an integer.
5. Write a program to take a number representing a month and print the name of the month using switch case.
6. Write a program that calculates the grade of a student based on their marks in a subject using nested if-else statements. Also print the range of marks for each grade using switch case.
7. Write a program to take a number as input and print all the even numbers up to that number using while and for loop.
8. Write a program to ask the user for an input to stop a loop or continue repeating after printing the iteration count using a do-while loop.

9. Write a program to find the maximum, minimum, sum and average of *n* numbers without using array.
10. Write a program that takes two integers as input and finds their greatest common divisor (GCD) using nested while loops and if statements.
11. Write a program that calculates the sum of the first *n* terms of the Fibonacci sequence, where *n* is entered by the user, using a for-loop.
12. Write a program that takes an integer as input and checks if it is a prime number.
13. Write a program that calculates the sum of the first *n* terms of an arithmetic series, where *n*, the first term and common difference of the series are entered by the user.
14. Write a program to compute the sum of the first *n* terms of the following series S =1-2+3-4+5……
15. Write a program to create an array with inputs from the user and print the same.
16. Write a program to perform following actions on an array entered by the user:
    a) Print the even-valued elements
    b) Print the odd-valued elements
    c) Print the array in reverse order
17. Write a program to take a matrix from the user and print the transpose of the same.
18. Write a program to ask for the name of the user and print the same.
19. Write a program to take a string of length more than 10 and find the number of vowels in the string. Also print the position of the vowels in the string.
20. Write a program using pointers to copy a string to another string variable without using library function.
21. Write a program that swaps two numbers using pointers.
22. Write a program to calculate Factorial of a number (i) using recursion, (ii) using iteration
23. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
24. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
25. Write a function to accept two arrays as argument and returns their sum as an array.
26. Write a program to implement struct in C. Create a structure of Student with RNo, Name and other credentials with proper datatype and print the same.
27. Write a program to implement union in C. Create a structure of Person with Pid, Name and other credentials with proper datatype and print the same.
28. Write a C program that opens a file for reading and displays the contents of the file in binary mode and text mode.
29. Write a C program that opens a file for reading and displays the contents of the file line by line on the screen.
30. Write a C program that opens a file in append mode and allows the user to add text to the end of the file.

Particulars of course designer:

Name : Risheraj Baruah
Contact No. : +91
8486942427
Email id : rishirajbaruah@gauhati.ac.in

Computer Fundamentals and Application Software (SEC)

1. **Learning Outcome:** At the end of the course, students will be able to:
   - Understand the basics of Computer System
   - Understand the basic of Software
   - Work with Word Processing Software
   - Work with a Spreadsheet Software
   - Work with a Presentation Software
2. **Prerequisite:** NIL
3. **Semester: 1**
4. **Course Type: SEC**
5. **Course Level: 100-199**
6. **Theory Credit: 1**
7. **Practical Credit: 2**
8. **Number of required hours:**
   a) Theory: 15 hrs
   b) Practical: 60 hrs
   c) Non Contact: 5 hrs

9. **List of reference books:**
   a) V. Rajaraman, "Fundamentals of Computer", 4th Ed., PHI, 2006
   b) R. Thareja, "Computer Fundamentals & Programming in C", Oxford University Press, 2013.

10. **Detailed Syllabus:**

Theory

Unit 1: Computer Fundamentals (8 hours)
Computer, Basic components of computer: Input unit, Output Unit and CPU, Memory – primary and secondary memory. Storage devices – magnetic storage devices: Magnetic Tape, Hard Disk, optical storage devices: CD, DVD, Input devices– mouse, keyboard, output devices – CRT and LCD monitors, printers: dot matrix printers, ink jet printers, laser printers.

Unit 2: Programming Basics (7 hours)
Introduction to programming languages. Low-level and high-level language and their characteristics. Compiler vs. interpreter. Software, application software, system software. Word processing software, Spreadsheet software, Presentation Software, Operating systems, functions of operating system, Open source software.

Laboratory

Unit 3: Word Processing Software (20 hours)
Opening and saving documents, Changing document views, Formatting text, Cutting, copying, and moving text, Finding and replacing text, Inserting special characters, Using AutoCorrect, Formatting paragraphs, Format painter, Pasting unformatted text, Creating and formatting tables, Creating and Formatting lists, Creating page headers and footers, displaying page numbers, Changing page margins, Adding a custom watermark to the page background, Adding page break, Adding comments to a document, Creating a table of contents,Working with images, Using Mail Merge, Macros, Linking and cross-referencing within a document, Using master documents, Shortcuts

Unit 4: Spreadsheet Software (30 hours)

Spreadsheets, sheets, and cells, Opening and saving spreadsheet files, Cell navigation, Sheet navigation, Inserting new sheets, Moving and copying sheets, Renaming sheets, Deleting sheets, Changing document view, Freezing rows and columns, Splitting the screen, Insert Date and time in a cell, auto fill, Sharing content between sheets, Validating cell contents, Replacing data, Multiple lines of text and wrapping, Manual line breaks, Merging cells, Splitting cells, Formatting texts and number in cells, Setting cell borders, setting cell background, AutoFormatting of cells, Conditional formatting, Filtering data, Sorting records, Cell comments, Using formula, Functions: SUM, AVERAGE, MAX, MIN, COUNT,COUNTA, IF, AND, OR, CONCATENATE, LEFT, RIGHT, MID, TODAY, NOW, YEAR, MONTH, DAY,
VLOOKUP, INDEX, MATCH, Pivot table, Inserting a page break, Headers and footers, Shortcuts, Column charts, Bar charts. Pie charts, Area charts, Line charts, Scatter, Bubble charts, Net charts

Unit 5: Presentation Software (10 hours)

Opening and saving files, Document views, Inserting a slide, Duplicate slide, Slide layouts, Adding and formatting text and image, Vertical text, Creating bulleted and numbered lists, Inserting images, tables, charts, or media, Inserting tables, OLE, Creating master slide, applying master slides, Adding comments, Printing handouts, starting slide show, hiding slides in slide show, Custom slide show, Setting slide transition effects, Slide element animation effects, Presenter Console

Particulars of course designer:
Name: Dr. Hasin A Ahmed
Contact No.: 8011810533
Mail id: hasin@gauhati.ac.in

COM0200104: COMPUTER ORGANIZATION

1. **Learning Outcome:**

   Student will able to learn about the structure, function and characteristics of computer systems.
   Student will understand the design of the various functional units and components of computers.
   Student will identify the elements of modern instructions sets and their impact on processor design.
   Student will able to learn about the function of each element of a memory hierarchy.
   Student will able to learn about identify and compare different methods for computer I/O.
   Student will able to understand the basic architecture of 8085 microprocessor.

1. **COURSE OUTCOMES:**

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

CO1: Describe the basic functional block of a computer.
CO2: Discuss different ways to represent data in memory and arithmetic operations performed on the data.
CO3: Explain the basic instruction set architectures of computer.
CO4: Analyze the hierarchy of memory organization and Input-Output systems used in computer.
CO5: Understand the basic architecture of 8085 microprocessor.

2. **Prerequisite: NIL**
3. **Semester: 2**
4. **Course Type: Compulsory**
5. **Course Level: 100-199**
6. **Theory Credit:** 4
7. **Practical Credit:** 0
8. **Number of required hours:**
   a) Theory: 60 hrs
   b) Practical: 0 hrs
   c) Non Contact: 5 hrs

10. List of reference books:

a) M. Morris Mano, *Computer System Architecture*, PHI publication.
b) Hamachar, Vranesic and Zaky, *Computer Architecture*.
c) William Stallings, *Computer Organization and Architecture*; Pearson.
d) Ramesh Gaonkar, *Microprocessor Architecture, Programming, and Applications with the 8085*, 5th Edition.

11. Detailed Syllabus:

UNIT 1: Introduction and Digital Logic (10 Lectures)

Definitions of Computer Organization and Architecture, History of computer architecture, Basic functional blocks of a computer, Logic Gates, Boolean Algebra, Map Simplifications, Combinational Circuits, Flip-Flops, Sequential Circuits, Decoders, Multiplexers, Registers and Counters

UNIT 2: Data Representation (8 Lectures)

Number Systems and Conversion, Complements, Fixed Point Representation, Floating Point Representation, Error Detection Codes, Computer Arithmetic - Addition, Subtraction, Other Binary codes: BCD, Excess-3, Gray code**.**

UNIT 3 : Register Transfer and Microoperations (6 Lectures)

Introduction to Register Transfer Language, Register transfer, Bus and Memory transfers, Arithmetic micro-operation- Binary adder, Binary adder-subtractor, Binary incrementer, Arithmetic circuit, Logic micro-operation, Shift micro-operation, Arithmetic logic shift unit.

UNIT 4: Basic Computer Architecture and Design (10 Lectures)

Instruction Codes: Stored Program Organization, Data path in a CPU, Computer Instructions, Instruction Cycle, Memory-Reference Instructions, I/O Interrupt: Types of Interrupt, Interrupt Cycle, Control Unit: Operations of a control unit, Hardwired control unit, Micro-programmed control unit

UNIT 4: Central Processing Unit (10 Lectures)

Computer registers, Types of register- general purpose registers, special purpose registers, index registers, General register organization, Stack organization, Computer instructions: Operands, Instruction format- Three-address instructions, Two-address instructions, One-address instructions, Zero- address instructions, Addressing modes, Data Transfer and Manipulation: Data transfer instructions, Data manipulation instructions, Arithmetic instructions, Logical and Bit manipulation instructions, Shift instructions, Program Control: Status bit conditions, Conditional branch instructions, Subroutine call and return, CISC and RISC architectures.

UNIT 5: Memory Organization (6 Lectures)

Semiconductor memories, Memory cells - SRAM and DRAM, Concept of hierarchical memory organization, Cache memory unit - Concept of cache memory, Mapping methods, Organization of a cache memory unit, Cache replacement policies, Write policy, Concept of virtual memory.

UNIT 6: I/O Organization (6 Lectures)

Access of I/O devices, I/O ports, I/O Interface, Modes of Transfer - Program controlled I/O, Interrupt driven I/O, DMA controlled I/O, Priority Interrupts, Handling interrupts.

UNIT 7: Basics of Microprocessor (4 Lectures)

Introduction to microprocessors, Case Study: 8085 Microprocessor – operations, instructions and its addressing modes.

SEC0206203 HTML Programming

1. Learning Outcome: On successful completion of this course, the student should be able to:

1. Introduce the fundamentals of Internet, and the principles of web design.
2. Construct basic websites using HTML and Cascading Style Sheets.
3. Design dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.

2. COURSE OUTCOMES:

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

CO1: Explain the structure and purpose of an HTML document.
CO2: Develop a basic webpage using HTML elements and attributes.
CO3: Inspect and debug HTML code for structural correctness.
CO4: Design a multi-page website using HTML with proper elements.

3. Prerequisites: NIL
4. Semester: 2
5. Course Type: SEC
6. Course Level: 100-199
7. Theory Credit: 1
8. Practical Credit: 2
9. No of Hours:

a) Theory: 15 hrs
b) Practical: 30 hrs
c) Non Contact: 0 hrs

10. List of Books:

1. Virginia DeBolt, Integrated HTML and CSS A Smarter, Faster Way to Learn Wiley / Sybex , 2006
2. Cassidy Williams, Camryn Williams Introduction to HTML and CSS, O'Reilly, 2015

11. Contents of Syllabus:

UNIT 1: The Basics (3 Lectures)

Introduction to HTML, the Head, the Body, Colors, Attributes, Check box, Radio Button, Text, Text Area, Lists, ordered and unordered.

UNIT 2: HTML Formatting (3 Lectures)

New Paragraph, Line Break, Blank Space, Preformatted text, Div element Bold text, Important text, Italic text, Emphasized text, Marked text, Small text, Deleted text, Inserted text, Subscript text, Superscript text, HTML quotations, HTML Comments, HTML colors

UNIT 3: Links (3 Lectures)

Introduction, Relative Links, Absolute Links, Link Attributes, Using the ID Attribute to Link within a Document

UNIT 4: Images (2 Lectures)

Putting an Image on a Page, Using Images as Links, Putting an Image in the Background

UNIT 5: Tables (5 Lectures)

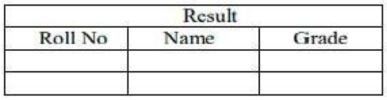
Creating a Table, Table Headers, Captions, Spanning Multiple Columns, Styling Table

UNIT 6: Forms (4 Lectures)

Basic Input and Attributes, Other Kinds of Inputs, Styling forms with CSS, Whereto Go fromHere.Form validation using JavaScript.

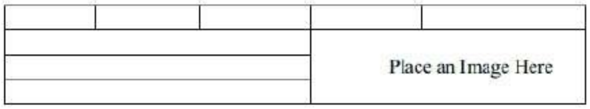
Practical/Lab work to be performed:

1. Create an HTML document with the following formatting options: (i) Bold (ii) Italics (iii) Underline (iv) Headings (Using H1 to H6 heading styles) (v) Font (Type, Sizeand Color) (vi) Background (Colored background/Image in background) (vii) Paragraph (viii) Line Break (ix) Horizontal Rule (x) Pre tag.
2. Create an HTML document which consists of: (i) Ordered List (ii) Unordered List (iii) Nested List
3. Create an HTML document which implements Internal linking as well as External Linking.
4. Create a table using HTML which consists of columns for Roll No., Student‘s name and grade



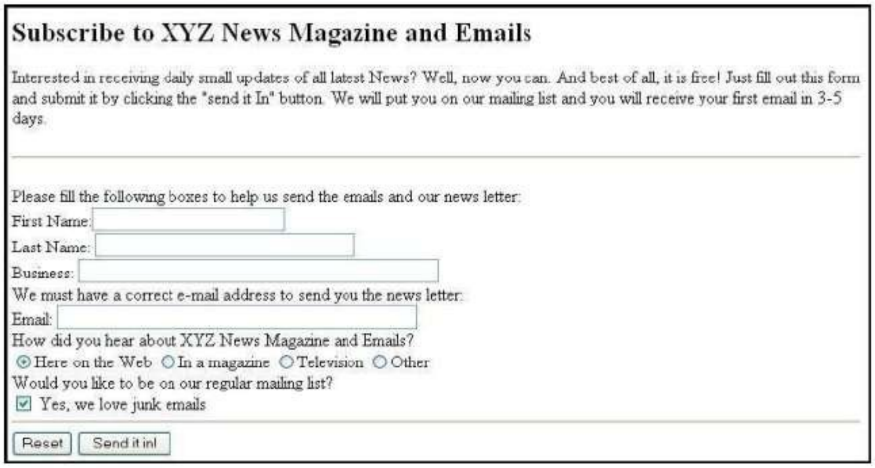
<table>
<tr><th colspan="3">Result</th></tr>
<tr><th>Roll No</th><th>Name</th><th>Grade</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

5. Create a Table with the following view:



<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="3"></td><td colspan="2" rowspan="3">Place an Image Here</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3"></td></tr>
</table>

6. . Create a form using HTML which has the following types of controls: (i) Text Box (ii) Option/radio buttons (iii) Check boxes (iv) Reset and Submit buttons.



Subscribe to XYZ News Magazine and Emails

Interested in receiving daily small updates of all latest News? Well, now you can. And best of all, it is free! Just fill out this form and submit it by clicking the "send it In" button. We will put you on our mailing list and you will receive your first email in 3-5 days.

Please fill the following boxes to help us send the emails and our news letter:

First Name:

Last Name:

Business:

We must have a correct e-mail address to send you the news letter:

Email:

How did you hear about XYZ News Magazine and Emails?

● Here on the Web ○ In a magazine ○ Television ○ Other

Would you like to be on our regular mailing list?

☑ Yes, we love junk emails

COM0300104: Object Oriented Programming using C++

2. **Learning Outcomes:** After successful completion of this course, students will be able to:
    - Differentiate between Structured programming and Object-Oriented Programming.
    - Learn the concept of objects and develop the ability of imagining real life concepts as objects and derive their properties and functions to operate these objects.
    - Develop programs using different object- oriented programming features such as data abstraction, polymorphism, inheritance, exception handling etc.

3. **COURSEOUTCOMES:**

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

   CO1: Define OOP and describe basic data types, different operators, control structures, Function prototyping and call by reference method used in C++.

   CO2: Discuss about the concept of class, object, memory allocation for objects, concepts of constructors and destructors used in C++.

   CO3: Explain the concept of function overloading and operator overloading.

   CO4: Illustrate the concept of inheritance, polymorphism and exception handling mechanism.

   CO5: Design C++ program to illustrate different concepts like class, object, constructor, inheritance, virtual functions used in OOP.

4. **Prerequisites: NIL**
5. **Semester: 3**
6. **Course Type:** Compulsory
7. **Course Level:** 200-299
8. **Theory Credit:** 3
9. **Practical Credit: 1**
10. **No of required hours:**

    a) Theory: 45 hrs

    b) Practical: 30 hrs

    c) Non Contact: 5 hrs
11. **List of Reference Books:**

    a) M. T. Somashekara, D. S. Guru et-al; *Object-Oriented Programming with C++, 2nd Edition*, PHI,2012.

    b) Bjarne Stroustrup, The*C++ Programming Language, Special Edition,* Pearson Education, 2004.

    c) Deitel&Deitel, *C++ How to program,* Pearson Education Asia, 6th Edition, 2008

    d) Schildt Herbert, *The Complete Reference C++,* Tata McGraw Hill, 4th Edition, 2003.

12. Contents of Syllabus:

A. Theory

UNIT 1: Introduction to object-oriented programming (3 Lectures)

Basic Concepts of Object-Oriented Programming and design, Benefits and applications of OOP.

UNIT 2: Introduction to C++ (6 lectures)

Structure of a Simple C++ program, Output operator, Input operator, Cascading of I/O operators, Tokens- keyword, identifiers, constants, strings and operators. Basic data types, User defined data types, Dynamic initialization of variables, Reference variables, Operators in C++, Scope resolution operator & applications, Member dereferencing operators, Memory Management operators, new and delete, Control Structures-simple if, if else, nested if, switch, while do, break and continue statements, Introduction to Functions-Function Prototyping, Call-by-reference, Return by reference, Inline functions, Default arguments, Const arguments.

UNIT 3: Classes and objects (11 Lectures)

Introduction - Defining a class; class versus structures, creating objects, accessing class members, defining member functions- outside the class definition and inside the class definition, outside functions as inline. Nesting of member functions, private member functions, memory allocation for objects. Array-declaring an array, accessing elements of an array, array of objects. Friendly functions. Basic Concepts of constructors and destructors with examples. Default constructor, Parameterized constructor, multiple constructors in a class. Constructor with default arguments, Copy constructor. Dynamic initialization of objects. Dynamic constructors and destructors.

UNIT 4: Function and operator overloading (10 Lectures)

Concept of Overloading. Function Overloading: Functions with different sets of parameters, default and constant parameters, Rules for overloading operators, defining operator overloading. Overloading unary operators -prefix and postfix operators. Overloading Binary operators and relational operators, Overloading using friend functions.

UNIT 5: Inheritance (12 Lectures)

Concept of Inheritance -defining derived classes. Types of inheritances, Making a private member inheritable, multilevel inheritance, multiple inheritance, Hierarchical inheritance, Hybrid inheritance, Virtual base classes, Abstract classes, Constructors in derived classes, nesting of classes, polymorphism-Compile time and Runtime polymorphism, Pointers to objects, “this” pointer, Pointer to derived classes, Virtual functions, Rules for virtual functions, Pure virtual functions.

UNIT 6: Exception handling (3 lectures)

Examples of exceptions and handling exceptions using try, catch and throw statements.

B. Practical’s

Following Practical / Lab works to be performed preferably in Linux Environment

1. Define a class named “triangle” to represent a triangle using the lengths of the three sides. Write a constructor to initialize objects of this class, given the lengths of the sides. Also write member functions to check

   (a) if a triangle is isosceles

   (b) if a triangle is equilateral

   Write a main function to test your functions.

2. Define a structure “employee” with the following specifications.

   empno : integer

   ename : 20 characters

   basic, hra, da : float

   calculate() : a function to compute net pay as basic+hra+da with float return type.

   getdata() : a function to read values for *empno*, ename, basic, hra, da.

   dispdata() : a function to display all the data on the screen

   Write a main program to test the program.

3. Define a class “circle” to represent circles. Add a data member radius to store the radius of a circle. Write member functions *area*() and *perimeter*() to compute the area and perimeter of a circle.

4. Define a class “complex” with two data members “real” and “imag” to represent real and imaginary parts of a complex number. Write member functions

   rpart() : to return the real part of a complex number

   ipart() : to return the imaginary part of a complex number

   add() : to add two complex numbers.

   mul() : to multiply two complex numbers.

   Write constructors with zero, one and two arguments to initialize objects.

5. Define a class “point” with two data members “*xordinate*” and “*yordinate*” to represent all points in the two-dimensional plane by storing their x co-ordinate and y co-ordinate values. Write member functions

   dist() : to return the distance of the point from the origin.

   slope(): to return the slope of the line obtained by joining this point with the origin.

   Write constructors with zero, one and two arguments to initialize objects. Also write a friend function to compute the distance between two points.

6. Define a class “string” with the following data members char *p; int size; and write member functions to do the following (without using library function) and using dynamic memory allocation.

- Length of the string
- Compare two strings
- Copy one string to another
- Reverse the string

Write suitable constructors and destructors. Also write a copy constructor for the class.

7. For the class “complex” defined in 4 above, overload the <<, >>, + and * operators in the usual sense. Also overload the unary – operator.

8. Define a class “time” to store time as hour, minute and second, all being integer values. Write member functions to display time in standard formats. Also overload the ++ and – operators to increase and decrease a given time by one second where the minute and hour values will have to be updated whenever necessary.

9. Define a class to store matrices. Write suitable friend functions to add and multiply two matrices.

10. Write a class-based program implementing static members.

11. Define a class student with the following specification:

rollno : integer sname : 20 characters

Derive two classes*artst* and *scst*. The class artst will represent students belonging to arts stream and the class scst will represent students belonging to science stream. The *artst* class will have additional data members ph, hs, en and as to store marks obtained by a student in three subjects Philosophy, History, English and Assamese. The class scst will have additional data member *sph*, *ch*, *ma* and *en* to store marks obtained in *Physics*, *Chemistry*, *Mathematics* and *English*.

Write the following member functions in the classes artst and scst; ctotal() : a function to calculate the total marks obtained by a student; *takedata*() : a function to accept values of the data members and show data() : a function to display the marks sheet of a student .

12. Define an abstract base class printer. Derive three classes’ laser-printer, line-printer and inkjet-printer. The derived classes will have data members to store the features of that particular printer. Write pure virtual function display() in the base class and redefine it in the derived classes.

13. Define a abstract base class figure and add to it pure virtual functions

display() : to display a figure
get() : to input parameters of the figure
area() : to compute the area of a figure
perimeter() : to compute the perimeter of a figure.

Derive three classes circle, rectangle and triangle from it. A circle is to be represented by its radius, rectangle by its length and breadth and triangle by the lengths of its sides. Write again function and write necessary statements to achieve run time polymorphism.

14. Write an interactive program to compute square root of a number. The input value must be tested for validity. If it is negative, the user defined function *my_sqrt*() should raise an exception.

COM0300204: Data Structure

1. Learning Outcomes: At the end of the course, students will be able to:

- Understand and apply the fundamental data structures and algorithms – such as arrays, linked lists, stacks, queues, trees, sorting and searching algorithms using C programming language.
- Analyze the time and space complexity of different algorithms and choose the appropriate algorithm for a given problem.
- Develop efficient algorithms to solve various computational problems by utilizing data structures and algorithms covered in the course.

1. COURSEOUTCOMES:

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

CO1: Distinguish between linear and non linear data structures.

CO2: Explain different operations performed on array, link list, stack, queue and binary tree.

CO2: Discuss different operations performed on binary tree, representation of binary tree using array and link list...

CO3: Explain about different sorting and searching algorithms applied on data.

CO4: Analyze the time complexity and space complexity of algorithm.

CO5: Design C++ program to show the operations performed on different data structure.

3. **Prerequisites**: NIL
4. Semester: 3
5. Course Type: Elective
6. Course Level: 200-299
7. Theory Credit: 3
8. Practical Credit: 1
9. No of required hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

10. List of Reference Books:

e) Weiss, Mark Allen. “Data Structures and Algorithm Analysis in C”. 3rd ed., Pearson, 2012
f) Sedgewick, Robert. “Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms”. 3rd ed., Addison-Wesley Professional, 2002.
g) Goodrich, Michael T., and Roberto Tamassia. “Data Structures and Algorithms in C”. 2nd ed., Wiley, 2011.
h) Gilberg, Richard F., and Behrouz A. Forouzan. “Data Structures: A Pseudocode Approach with C”. Narosa Publishing House, 2009.

11. Contents of Syllabus:

A. Theory

Unit 1: Data Structures Overview and Arrays (8 Lectures)

Concepts of Data Types, Abstract Data Type, Data Structure, Fundamental and Derived Data Types. Importance of data structures. Array as a data structure (characteristics, advantages, disadvantages). Representation of arrays – single and multidimensional. Address calculation of array element using column and row major ordering. Address translation functions for one & two dimensional arrays. Insertion and deletion in arrays. Use of arrays for large number representation.

Unit 2: Linked Lists (9 Lectures)

Initialization and implementation of structures. Structure and pointers. Self referential structure. Introduction to linked lists. Singly linked list, doubly linked list, circular linked list. Operations on lists – creation, insertion, deletion, traversal, merging and splitting.

Unit 3: Stacks and Queues (9 Lectures)

Definition of Stack and Queue. Representation of stacks and queues using arrays and linked lists. Stack operations – push, pop. Queue operation – enqueue, dequeue. Circular Queue, Priority Queue, Conversion of infix arithmetic expression containing arithmetic operators and parenthesis to postfix and prefix expression. Evaluation of postfix expression.

Unit 4: Binary Trees (8 Lectures)

Definition of Trees – General tree and Binary tree. Basic terminologies – parent, child, height, depth, leaf, node, internal nodes, external nodes. Brief concept of Forest, ordered trees, strictly binary tree, complete binary tree. Representation of trees using arrays and linked lists. Binary tree traversal methods – pre-order, in-order, post-order. Recursive and non-recursive algorithms for traversal methods. Binary search trees. Operation on BST – creation, insertion and deletion of a node. Definition and characteristics of threaded binary trees. Min heap and Max heap.

Unit 5: Searching and Sorting (6 Lectures)

Linear and binary search. Indexed search. Hashing. Hash Functions – division method, mid square method, folding. Conflict resolution – linear and quadratic probe. Sorting algorithms – Insertion sort, Selection sort, Bubble sort, Merge sort, Quick sort, Counting sort, Heap sort. In-place sorting and stable sorting.

Unit 6: Analysis of Algorithm and Complexity (5 Lectures)

Complexity measures of an algorithm – Time and space complexity. Average case and worst case analysis. Asymptotic notation as a measure of algorithm complexity, O and θ notations. Analysis of sorting algorithms and Searching algorithms in terms of time and space complexity in best, average and worst case.

Time and Space complexity of algorithms, average case and worst case analysis, asymptotic notation as a measure of algorithm complexity, Θ and O notation. Analysis of sorting algorithms- Selection sort, Bubble sort, Insertion sort, Heap sort, Quick sort and analysis of searching algorithms – linear search and binary search.

List of Practical

(This is a suggestive list only. Questions need not be restricted to this list. The practical are advised to be performed in Linux environment using C programming language.)

1. Write a program to declare an array and initialize the values according to the user. Now ask the user for a number *n* and return the *n*th element from the array.
2. Write a program to implement array initialized with the numbers divisible by three up to 30. Write a function which accepts the array and return the positions of the even numbers in the array.
3. Implement linked list in a program by writing functions for the following:
    a. Create a singly linked list of *n* nodes
    b. Count the number of nodes in the list
    c. Print the values of all the nodes
    d. Add a node at first, last and *k*th position in the linked list
    e. Delete a node from first, last and *k*th position
    f. Search for an element in the list. If found, return the position of the node. If not found, return a negative value.
4. Write a program to implement doubly linked list.
5. Write a function to concatenate two linked lists.
6. Write a program to take a number *k* and split the linked list after *k*th position.
7. Write a program to merge two sorted linked lists.
8. Write a program to implement list of lists.
9. Write a program to implement stack using array. Use push and pop operations on the array representation of the stack. Check whether the stack is full or empty.
10. Write a program to implement stack using linked list. Use push and pop operations on the stack by inserting nodes and deleting nodes from the linked list. Also check if the stack is full or empty.
11. Write a program to evaluate a simple postfix expression using stack.
12. Write a program to convert a decimal number into binary number using stack.
13. Write a program to implement queue using array. Add new elements to the queue and remove elements from the queue represented by array. Check whether the queue is full or empty.
14. Write a program to implement queue using linked list. Add new elements to the queue and remove elements from the queue represented by linked list. Also check whether the queue is full or empty.
15. Implement binary search and linear search algorithms on arrays.
16. Implement binary search tree using array by writing a program to:
    a. Create a binary search tree using array
    b. Print the prefix notation of the BST

   c. Print the infix notation of the BST
   d. Print the postfix notation of the BST
   e. Search for an element in the BST

17. Implement binary search tree using linked list by writing a program to:
    a. Create a binary search tree using linked list
    b. Print the prefix notation of the BST
    c. Print the infix notation of the BST
    d. Print the postfix notation of the BST
    e. Search for an element in the BST
18. Implement following sorting algorithms:

    Bubble sort, Insertion sort ,Selection sort, Counting sort

COM0400204: Database Management System

1. **Learning Outcome:** On successful completion, the student should be able to:
   - Learn database concepts and its architectural components.
   - Describe different data models used for designing a database.
   - To create a database using relational models and entity relationships concepts
   - Normalize a database into various normal forms
   - Design SQL queries to handle a relational database.

2. **COURSEOUTCOMES:** At the end of the course, students will be able to:
   CO 1: Describe database management system architectures, its types, advantages and disadvantages.
   CO 2: Apply Entity Relationship modeling to real world problems and mapping ER to relational model.
   CO 3: Interpret functional dependencies and categorize a table to specific normalization.
   CO 4: Analyze various concurrency problems that may happen in database design process.
   CO 5: Construct complex SQL queries to fetch, delete, update, and create data in DBMS software.
3. **Prerequisite:** NIL
4. **Semester: 4**
5. **Course Type: Compulsory**
6. **Course Level: 200-299**
7. **Theory Credit:** 3
8. **Practical Credit:** 1
9. **Number of required hours:**
   a) Theory: 45 hrs
   b) Practical: 30 hrs
   c) Non Contact: 5 hrs

10. List of reference books:

a) Dr. Satinder Bal Gupta and Aditya Mittal, *Introduction to Database Management System¸* University Science Press
b) A. Silberschatz, H.F. Korth, S. Sudarshan, *Database System Concepts*, McGraw Hill
c) R. Elmasri, S.B. Navathe, *Fundamentals of Database Systems*, Pearson Education
d) Dr. Rajive Chopra, *Database Management System (DBMS): A Practical Approach*, S. Chand Publication

11. Detailed Syllabus:

UNIT-1: **Introduction to Database Management Systems (5 Lectures)**

Basic Definition and Concepts: *Data, Information, Meta Data, Data Dictionary, Database, Fields*, *Records* and *Files*. Definition of Database Management System (DBMS), Primary Functions of DBMS, Traditional File approach, Traditional file approach versus database management system approach, Disadvantages of Traditional File System, Need of a DBMS, Components of a DBMS, Advantages of DBMS, Disadvantages of Database Systems, Various uses of database System Applications, Database Users: *End users or naive users*, *Onlineusers*, *Application Programmers*, *Database Administrator* (*DBA*), Responsibilities of DBA.

UNIT 2: Database Management System Architecture (6 Lectures)

Definition of *Schemas*, *sub-schema* and *Instances*. Data Independence: *Physical Data Independence* and *Logical data Independence*. Three-tier architecture of DBMS, Advantages of three-level Architecture, basic concept of data model, Characteristics of Data Models, Types of Data models: *Record Based Data Models*, *Object Based Data Model* and *Physical Data Models*. Relational Data Model, Types of database Systems: *Single-user* database systems, *Multiuser* database systems, *Centralized* database systems, *Distributed* database systems and *Client/Server* database systems.

UNIT 3: E-R Modeling (8 Lectures)

Basic Concepts: *Entity*, *Attributes*, *Entity Sets*, *Domain*. Types of attributes: *Simple and Composite Attributes*, *Single Valued and Multi-valued Attributes*, *Derived Attributes and Stored Attributes*. Types of Entity Sets*: Strong Entity Sets* and *Weak Entity Sets.* Concept of Relationship and Relationship sets, Types of Relationship: One-to-One, One-to-Many, Many-to-One and Many-to Many, Various Symbols used in ER Diagram, Mapping constraints: *Mapping Cardinalities (Cardinality Ratios)* and *Participation Constraints*. Definition of Key, Types of Keys: *Super Key*, *Candidate Key, Primary Key, Alternate Key* and *Foreign Key*. Symbols used in E-R diagrams, Conversion of an ER and Diagram in to Relational Tables

UNIT 4: Relational Model and Relational Algebra (7 Lectures)

Definition of Relation, Data Structure of Relational Database: *Relation*, *Tuples*, *Attributes Domain, Degree* and *Cardinality.* Integrity Constraints, Domain Constraints, Key Constraints, Advantages and Disadvantages of Relational Model, Relational, Definition of Relational algebra, Operations in Relational Algebra: *Selection*, *Projection, Division, Rename, Union, Intersection, Set Difference, Natural-join operation, Outer join, Inner Join, Cartesian Product* and *Assignment operation*. Aggregate Functions and Operations: *Average, Maximum, Minimum, Sum* and *Count*.

UNIT 5: Functional Dependency and Normalization (8 Lectures)

Definition of Functional Dependency, Armstrong's Axioms in Functional Dependency, Types of Functional Dependency: *Partial Dependency, Full Functional Dependency, Transitive and Non-transitive Functional Dependency*, Armstrong's Axiom, Closure of a set of Functional Dependency, Closure of an Attribute, Definition of Canonical Cover, Algorithm to find the canonical cover of a FD set, Anomalies in relational database*:* *Insertion*, *Deletion* and *Update* anomalies, Concepts of Normalization, Benefits of Normalization, Types of Normal Forms: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF) and Boyce–Codd Normal Form (BCNF)

UNIT 6: Transaction and Concurrency Control (4 Lectures)

Definition of Transaction, ACID Properties of transaction, Transaction States, Definition of Concurrency Control, Need of Concurrency Control, The Lost Update Problem, The Uncommitted Dependency Problem, The Inconsistent Analysis Problem, Serializability: *View Serializability* and *Conflict Serializability*

UNIT 7: SQL Queries (7 Lectures)

Database Languages (Data Definition Languages, Data Manipulation Languages), Characteristics of SQL, Basic data types in SQL, Data-definition language (DDL) commands: *Create Database, Create Table*, *Drop Table*, *Alter Table*. SQL Constraints: *Primary Key, Foreign Key, Not Null, Unique, Check, Default,* .Data Manipulation Language (DML) commands: *Insert Into*, Delete, *Select*, *Update*. SQL clauses: *Where, Order By, Having, Group By* and *Like*. SQL join operations: *Inner Join, Left Outer Join, Right Outer Join* and *Full Join*. SQL aggregate functions: *sum(), count(), max(), min()* and *avg()*

Lab Contents:

Practical / Lab work to be performed:

- Implementation of SQL DDL statements in MySQL DBMS: CREATE DATABASE, CREATE TABLE, ALTER TABLE, RENAME, DROP DATABASE/TABLE
- Use of SQL DML statements in MySQL DBMS: INSERT, SELECT, UPDATE, DELETE SQL commands
- Implementing following constraints in MySQL DBMS: PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE and DEFAULT
- Handling following SQL clauses in MySQL DBMS: WHERE, GROUP BY, ORDER BY, HAVING, IN, BETWEEN, LIKE
- Working with following aggregate functions in MySQL DBMS: COUNT, AVG, MAX, MIN and SUM
- Working with transaction processing command in MySQL DBMS: START TRANSACTION, COMMIT and ROLLBACK Statements, SET auto commit

COM0400304: Mathematical Foundation of Computer Science

1. Learning Outcome: After successful completion of this course, students will be able to:

- Learn the concepts of set, relation, and function from Computer Science point of view.
- Understand the basic idea of counting and use it in counting under various constraints.
- Understand graphs and its different representations in Computers. How to model real life problems using graphs. Learn a few basic graph traversal algorithms.
- Understand Mathematical Logic from algorithmic point of view.

2. COURSEOUTCOMES:

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

CO 1: Define and describe sets, relations, functions, its properties and operations.
CO 2: Associate with the principles of counting, inclusion, induction and its applications.
CO 3: Analyze the growth of functions and discuss asymptotic notations.
CO 4: Distinguish and create different graphs, discuss its properties and applications.
CO 5: Solve problems that are revolving around mathematical logic e.g. connectives, truth, tables, tautologies and contradictions.

3. Prerequisites: Nil
4. Semester: 4
5. Course Type: Elective
6. Course Level: 200-299
7. Theory Credit: 4
8. Practical Credit: 0
9. No of Hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) *Elements of Discrete mathematics*, C.L. Liu , D.P. Mahopatra; 2nd Edition , Tata McGraw Hill, 1985,
b) **Discrete Mathematics and Its Applications**, Kenneth Rosen, Sixth Edition, McGraw Hill 2006.
c) *Introduction to Algorithms*, T.H. Coremen, C.E. Leiserson, R. L. Rivest; 3rd edition Prentice Hall of India, 2009.
d) *Discrete Mathematics and Graph Theory*; Grimaldi, 5th Edition; 2019, Pearson.

11. Detailed Syllabus:

A. Theory

UNIT 1: (16 Lectures)

Sets, Relations and Functions

Sets: definition of set, cardinality of sets, finite, countable and infinite sets. Operations on sets, Venn diagram. Principle of inclusion and exclusion and their applications on simple problems. Multisets.

Relations: Definition and properties of binary relations, closures of relations, equivalence relations, equivalence classes and partitions, n-ary relations and representation of n-ary relations as tables. Partial ordering relations and lattices,

Functions: Definition of function**,** one-to-one and onto, principles of mathematical induction. Concave and convex functions.

UNIT 2: Combinatorics

(15lectures)

Basic of counting principles, principle of inclusion-exclusion, application of inclusion and exclusion, Mathematical Induction. Pigeonhole principle, generalized Pigeonhole principle and its application, permutations and combinations, circular permutations, permutations with repetitions, combinations with repetitions, permutations of sets with indistinguishable objects

UNIT 3: Growth of Function (5 Lectures)

Asymptotic behavior of functions, Asymptotic Notations - Big-O and Theta. Summation formulas and properties, Bounding Summations.

UNIT 4: Graph Theory (12 Lectures)

Basic Definition of graph, Directed, Undirected and Weighted Graphs. Representation of graphs in Computers – Adjacency Matrix and Adjacency Lists. Degree of vertices – in degree and out degree. Paths, Cycles and Acyclic graphs. Simple operations on graphs and amount of computations required for each operation. Connected graph, Tree and Forest. Bipartite graph, Algorithms on graph traversals- Breadth first search, Depth first search.

UNIT 5: Mathematical Logic (12 Lectures)

Connectives, truth tables, Tautologies and Contradictions, Equivalence and Implications, NAND and NOR, Normal forms- CNF, DNF, Converting expressions to CNF and DNF, Theory of inference, Propositional Calculus, Predicate calculus (only introduction), predicates and quantifiers.

COM0400404: Operating System

1. **Learning Outcomes**: After successful completion of this course, students will be able to:
   - Learning Outcomes: After completing this course, students will have understanding of the internal structure and usage of various components related to an operating system.

2. COURSEOUTCOMES:

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

CO1: Identify different operating systems and its types, functions, applications.
CO2: Explain process, threads and solve process scheduling problems.
CO3: Compare solutions of process synchronizations methods.
CO4: Discuss process deadlock handling techniques.
CO5: Solve memory management issues and discuss memory management solutions.

3. **Prerequisites**: NIL
4. Semester: 4
5. Course Type: Elective
6. Course Level: 200-299
7. Theory Credit: 3
9. Practical Credit: 1
10. No of required hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

11. List of Reference Books:

i) Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne,Wiley
j) Modern Operating Systems, Andrew S. Tanenbaum, Prentice-Hall Of India Pvt. Limited

12. Contents of Syllabus:

A. Theory

Unit I: Introduction **(7 hrs)**

Application vs system software, operating system as system software, operating structure structure, types of operating systems: batch operating system, multiprogramming operating system, multitasking operating system, distributed operating system, real time operating system, multi user operating system, major functions of operating system: Process Management, Process Synchronization, Memory Management, CPU Scheduling, File Management, I/O Management, Security, virtualization, cloud computing, open source operating system, history of operating system, the shell, system call, system boot

Unit II: Process and threads (10 hrs)

Process, process states: new, running, waiting, ready and terminated, Process Control Block (PCB),information stored in PCB, scheduling queue: job queue, ready queue and device queue, schedulers: long term schedulers, medium term scheduler and long term scheduler, swapping, degree of multiprogramming, I/O-bound and CPU-bound processes, context switching, inter-process communication: shared memory systems and message passing systems, socket, remote procedure call, threads, user threads, kernel threads, multi threading models: Many-to-One Model, One-to-One Model, Many-to-Many Model, CPU scheduling, Scheduling Criteria, scheduling algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling

Unit III: Process synchronization (8 hrs)

Race condition, critical section problem, Peterson's algorithm, Bakery algorithm, synchronization hardware: locking, synchronization software tools: mutex lock, semaphore (counting and binary), semaphore implementation, classic synchronization problems: bounded buffer problem, the readers–writers Problem, the dining-philosophers problem, monitor, synchronization in windows, synchronization in Linux

Unit IV: Deadlock (10 hrs)

Deadlock, operations of a process performs while using a resource: Request. Use and Release, physical and logical resources, Necessary conditions: mutual exclusion, hold & wait, no preemption and circular wait, resource allocation graph, deadlock prevention: definition, preventing mutual exclusion, preventing hold & wait, preventing no preemption and preventing circular wait, deadlock avoidance: definition, safe state, safe sequence, resource allocation graph based algorithm and Banker's algorithm, deadlock detection: definition, wait-for graph, algorithm to detect deadlock for single instance resources, algorithm to detect deadlock for multiple instance resources and recovery from deadlock: process termination and resource preemption

Unit V: Memory Management (10 hrs)

Memory hierarchy, base register, limit register, address binding, logical and physical address spaces, memory management unit, relocation register, swapping, contiguous memory allocation: definition, memory protection, fixed partition scheme, variable partition scheme, first-fit, best-fit &worst-fit allocation strategies, non-contiguous memory allocation: simple paging and simple segmentation, internal and external fragmentation, TLB, virtual memory, demand paging, page fault, locality of reference principle, performance of demand paging, page replacement algorithms: FIFO, Optimal and LRU, allocation of frames: equal allocation and proportional allocation, global and local page replacement algorithms, thrashing

Practicals:

1. Basic linux commands: pwd, ls, cd, mkdir, rmdir, rm, touch, man, cp, mv, locate, head, tail Advanced commands: echo, cat, sudo, df, tar, apt-get, chmod, hostname,

useradd, passwd, groupadd, grep, sed, uniq, wc, od, gzip, gunzip, find, date, cal, clear, top, ps, kill

2. Shell scripting in linux: shell, types of shell, shell script, echo command, shell variables,
3. special variables ($$, $0, $n, $#, $?, $!), array, assignment operator (=), equality operator (==), not equality operator (!=), arithmetic operators (+,-, *, /, %), comparison operators (-eq, -neq, -gt, -lt, - ge, -le), logical operators (!,-o, -a), if...else statement, case...esac statement, while loop, for loop, break statement, continue statement, shell functions 7 classes
4. Using system calls in C program in linux: fork(), exec(), exit(), getpid(), mkdir(), rmdir() etc.

COM0400504: Java Programming

1. **Learning Outcome:**
   After completing this course, students will be
   - Familiar with the core concepts of java programming and classes of swing package.

2. COURSEOUTCOMES:

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

CO 1: Describe java programming constructs and operate different java IDEs.
CO 2: Explain working principle of various building blocks of java platform.
CO 3: Apply object oriented approach using java language to solve required programming problems.
CO 4: Differentiate different java string, array methods and its usage.
CO 5: Design and develop GUI applications using advanced java packages.

3. Prerequisites: NIL
4. Semester: 4
5. Course Type: Elective
6. Course Level: 200-299
7. Theory Credit: 3
8. Practical Credit: 1
9. No of Hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) *Java: The Complete Reference*, Herbert Schildt, McGrawHill
b) *Java How to Program,* Paul Deitel, Harvey Deitel, Pearson

11. Contents of Syllabus:
A. Theory
Detailed Syllabus:
Unit I: Introduction (3 hrs)

High level language, compiled and interpreted languages, history of java programming language, compilation of java code, bytecode, java interpreter, javac and java command, path environmental variable, Java IDE, features of java programming language: simple, object oriented, robust, architecture neutral and interpreted

Unit II: Data types, operators and control statements (12 hrs)

Java as strongly typed language, primitive data types, integer data types: byte, short, int and long, floating point data types: float and double, character data type, boolean data type, literals: integer literals, floating-point literals, boolean literals, character literals and string literals, declaring a variable, dynamic Initialization, the scope and lifetime of variables, typecasting in java, one dimensional array, multi dimensional array, arithmetic operators: the basic arithmetic operators, the modulus operator, arithmetic compound assignment operators, increment operator and decrement operator, bitwise operators, relational operators, short circuit logical operator, the assignment operator, branching statements: if-else and switch-case statements, looping statements: while, do-while, for and for-each statements, jump statements: break and continue

Unit III: Object oriented features of java (10 hrs)

Defining a class, member variable and member methods, access specifies: default, private and public, declaring objects, assigning object reference variables, constructors, parameterized constructors, the this keyword, garbage collection, the finalize( ) method, overloading methods, overloading constructor, static keyword, final keyword, command line arguments in java, inheritance, super class and sub class, protected access specified, super keyword, constructor call in multilevel inheritance, method overriding, dynamic method dispatch, abstract class, interfaces, type wrappers

Unit IV: String handling and packages (5 hrs)

String class, String constructors, String length, special string operations: string literals, string concatenation, string concatenation with other data types, string conversion and toString( ), character extraction: charAt( ), getChars( ), string Comparison: equals( ) and equalsIgnoreCase(), regionMatches( ), startsWith( ) and endsWith( ), equals( ) Versus ==, compareTo( ), searching strings, data conversion using valueOf( ), StringBuffer, StringBuffer constructors, length( ) and capacity( ), ensureCapacity( ), setLength( ), charAt( ) and setCharAt( ), getChars( ), package, defining a package, CLASSPATH, importing packages

Unit V: Exception handling and I/O (5 hrs)

Exception-handling, exception types, uncaught exceptions, try and catch block, multiple catch blocks, nested try statements, throw, throws, finally, java's built-in exceptions, creating own exception classes, java I/O classes, reading console input, writing console output, reading and writing files

Unit VI: Swing package and database connectivity (10 hrs)

Swing package, simple GUI-Based Input/Output with JoptionPane, Jframe, JLabel, JTextField, Jbutton, handling event in a Jframe object, layout managers: BorderLayout, FlowLayout, GridLayout, CardLayout, GridBagLayout, JtoggleButton, JCheckBox, JRadioButton, Jlist, JcomboBox, JDBC, JDBC driver, connectivity steps, connectivity with MySQL, DriverManager class, Connection class, Statement class, Result Set class, Prepared Statement class

(b) Practical

- Java programs to demonstrate the use of data types and operators
- Java input through Scanner class and JOptionPane class
- Java programs to demonstrate the use of control statements.
- Java programs to demonstrate the use of classes, objects, visibility modes, constructors and destructor.

- Java programs to demonstrate the use of inheritance and polymorphism.
- Java programs to demonstrate the use of polymorphism.
- Java programs to handle strings, Java programs implementing exception handling.
- Demonstrating the use and creation of packages in java.
- Java program with JFrame, JTextfield and JButton with event handling
- Using JLabel, JTextArea and JPasswordField in java with event handling
- Working with layout managers in JFrame
- Using JCheckBox, JRadioButton and JComboBox in a JFrame
- Connecting JFrame components to a DBMS

COM0400604: Python Programming

1. Learning Outcome:

After completing this course, students

- Know about fundamentals of Python Programming and Problem Solving.

2. COURSEOUTCOMES:

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

CO1: Identify different keywords, operators and input/output formatting in Python.
CO2: Explain control statements and functions in Python
CO3: Apply object-oriented programming principles in Python.
CO4: Analyze the use cases of Python libraries and their significance in various applications.
CO5: Assess the robustness of Python code by applying exception handling and understanding the implications of database operations.

3. Prerequisites: NIL

4. Semester: 4

5. Course Type: Elective

6. Course Level: 200-299

7. Theory Credit: 3

8. Practical Credit: 1

9. No of Hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) *Core Python Programming*, R. Nageswara Rao, Dreamtech Press.
b) *Python: The Complete Reference*, Martin C. Brown, McGraw Hill Education.
c) http://docs.python.org/3/tutorial/index.html

11. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit 1: Introduction to Python Programming (7 hrs)

Introduction, Installation of Python Interpreter, Python Shell, Code Indentation, Identifiers and Keywords, Literals, Strings, Operators (Arithmetic, Relational, Logical, Assignment, Ternary, Bitwise, Increment and Decrement Operators), Input and output statements, Output Formatting.

Unit 2: Control Statements and Functions (7 hrs)

Branching, Looping, Conditional Statement, Exit Functions, Break, Continue, Pass, Defining Functions, Default Arguments. Scope of Functions, Function Documentation, Lambda Functions & Map.

Unit 3: Python Data Structures (6 hrs)

List (List, Nested List, List as Matrix), Tuple, Set, Dictionary.

Unit 4: Exception Handing (4 hrs)

Errors, Exception Handling with try, Multiple Exception Handling, Writing own Exception.

Unit 5: File Handling (6 hrs)

Understanding read function, read(), readline() and readlines(), Understanding write functions, write() and writelines(), Programming using file operations, Reading config files, Writing log files in python.

Unit 6: OOP in Python (3 hrs)

Creating Classes in Python, Instance Methods, Inheritance, Polymorphism, Exception Classes and Custom Exceptions.

Unit 7: Introduction to Libraries in Python (5 hrs)

NumPy, Matplotlib, OpenCV, Tkinter.

Unit 8: Python SQL Database Access (7 hrs)

Introduction to database driven program, Database Connection, Database Operations: INSERT, READ, UPDATE, DELETE, COMMIT AND ROLLBACK.

(b) Practical

- Introduction to Python console, operators, input and output statements.
- Python control statements and functions
- Data Structures in python
- Exception Handling
- File Handling
- Object Oriented Python programming
- Introduction to libraries (NumPy, Matplotlib, OpenCV)
- Python SQL Database Connection and database operations

COM0500104: Computer Networks

1. Learning Outcome: After completing this course, students

- Student will able to learn about the general principles of data communication.
- Student will able to learn about how computer networks are organized with the concept of layered approach.
- Student will able to learn about how signals are used to transfer data between nodes.
- Student will able to learn about how packets in the Internet are delivered.
- Student will able to learn about how routing protocols work.
- Student will able to learn about functions of transport layer
- Student will able to learn about functions of application layer

2. COURSEOUTCOMES:

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

CO1: Define internet and describe different network topologies and types of networks.
CO2: Discuss about different transmission mediums, encoding methods, transmissions modes and switching techniques.
CO3: Explain the concept of framing, error control, stop and wait protocols sliding window protocols, protocols of MAC sub layer.
CO4: Illustrate the concept of IP address, three way handshaking and DNS.
CO5: Compare different routing algorithms used by network layer.

3. Prerequisites: NIL
4. Semester: 5
5. Course Type: Elective
6. Course Level: 300-399
7. Theory Credit: 3
8. Practical Credit: 1
9. No of Hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

10. List of Books:

d) B. A. Forouzan: *Data Communications and Networking*, Fourth edition, THM, 2007.
e) A. S. Tanenbaum: *Computer Networks*, Fourth edition,PHI , 2002.

11. Contents of Syllabus:
A. Theory
Detailed Syllabus:
UNIT 1: Introduction to Computer Networks (5 Lectures)

Data communication system and its components, Definition of network, Types of network, Network topologies, Network protocol, Layered network architecture, Overview of OSI reference model, Overview of TCP/IP protocol suite.

UNIT 2: Physical Layer Communication (10 Lectures)

Analog and digital signal, Definition of bandwidth, Maximum data rate of a channel, Line encoding schemes, Transmission modes, Modulation techniques, Multiplexing techniques-FDM and TDM, Transmission media-Guided and Unguided, Switching techniques- Circuit switching, Packet switching, Connectionless datagram switching, Connection-oriented virtual circuit switching.

UNIT 3: Data Link Layer Functions and Protocol (10 Lectures)

Definition of Framing, Framing methods, Error detection techniques, Error correction techniques, Flow control mechanisms- Simplex protocol, Stop and Wait ARQ, Go-Back-N ARQ, Point to Point protocol.

UNIT 4: Multiple Access Protocol and Networks (5 Lectures)

Basics of ALOHA protocols, Basics of CSMA/CD protocols, Ethernet LANS, Connecting LAN and back-bone networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways

UNIT 5: Networks Layer Functions and Protocols (8 Lectures)

Connection oriented vs. Connectionless services, Definition of Routing, Routing algorithms, IP protocol, IP addresses, ARP, RARP

UNIT 6: Transport Layer Functions and Protocols (4 Lectures)

Transport services, TCP vs. UDP protocol, TCP connection establishment- Three way handshakes, TCP connection release

UNIT 7: Overview of Application Layer Protocols (3 Lectures)

Overview of DNS, Overview of WWW, URL, Email architecture, HTTP protocol

B. Practical / Lab work to be performed (15 Practical Classes)

- Implement the data link layer farming methods such as Bit Stuffing.
- Study of different types of Network cables.
- Study of network IP.
- Connect the computers in Local Area Network.
- Study of basic network command and Network configuration commands.
- Socket programming in C language.
- Configure a Network topology using packet tracer software.
- Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
- Simulate and implement Stop and Wait protocol for noisy channel.
- Simulate and implement Go-Back-N sliding window protocol.
- Simulate and implement Selective Repeat sliding window protocol.
- Simulate and implement Dijkstra Algorithm for shortest path routing.
- Simulate and implement Distance vector routing algorithm

COM0500204: Software Engineering

1. **Learning Outcome:** On successful completion of course, the student should be able to:
   - Determine the primary problems that impact all software development processes.
   - Choose relevant software development processes models, methodologies, and strategies for managing a specific software development process, and justify the choices
   - Implement different software estimation metrics such as cost, effort size, staffing etc.
   - Describe various software design approaches and various coding and testing strategies used in software engineering principles
   - Know about software reliability and how to calculate software maintenance cost

2. COURSEOUTCOMES:

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

CO1: Identify different software development processes and their challenges.
CO2: Explain software requires specification and translate it into an implementable design, following a structured and organized process.
CO3: Apply different software estimation metrics such as cost, effort size, staffing etc.
CO4: Design effective use of UML, along with design strategies such as defining software architecture, separation of concerns and design patterns.
CO5: Develop a system using OOP concepts

3. Prerequisites: NIL
4. Semester: 5
5. Course Type: Elective
6. Course Level: 300-399
7. Theory Credit: 4
8. Practical Credit: 0
9. No of Hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) Rajib Mall: *Fundamentals of Software Engineering*; PHI Learning Pvt. Ltd.
b) Roger S. Pressman: *Software Engineering: A practitioner's Approach*; McGraw Hill.

11. Contents of Syllabus:
A. Theory
Detailed Syllabus:
Unit 1: Introduction (4 Lectures)

Definition of Software Engineering, differentiation between Computer Science, Software Engineering and System Engineering, Program V/s software product, Exploratory style and modern style of software development, need of software engineering, characteristics of good software product

Unit 2: Software Development Life Cycle models (7 Lectures)

Definition of software development Life cycle (SDLC) models, Various life cycle modes: Classical Waterfall model, Iterative Waterfall model, Prototyping model, Evolutionary (Incremental) model, Spiral model, Agile Model, Agile V/s traditional SDLC Models, SCRUM model, Advantages and disadvantages of each of these SDLC models.

Unit 3: Requirement Analysis and Specification (7 Lectures)

What is Requirement Analysis and Gathering, Concept and Importance of Feasibility Study in Software design, Types of Feasibility: *Technical*, *Economical* and *Operational* feasibility, Software Requirement Specification (SRS) document, Components of an SRS (Software Requirement Specification): Functional and Non-Functional Component, Properties of a good SRS, Different users of SRS, Techniques to represent Complex Logic in SRS: Decision Tree and Decision Table.

Unit 4: Software Project Management (15 Lectures)

Basic idea of Software Project Management, Job Responsibilities of a Software Project Manager, Need of SPMP (Software Project Management Plan) document, Contents of SPMP, Need of Software documentation, Internal and External documentation, Software size estimation using Lines of Code (LOC), Merits and Demerits of LOC metric, Function Point Metric, 3D Function Point metrics, Project Estimation Techniques: *Empirical estimation* and *Heuristics estimation* techniques. Empirical estimation techniques: *Delphi Cost Estimation* and *Delphi Cost Estimation.* Heuristic Estimation Techniques: *Basic COCOMO model* and *Intermediate COCOMO model.* Project Scheduling: *Work break down structure, Activity Networks* and *Critical Path Method.* Project Team structure: *Chief Programmer team* and *Democratic team* structure.

Unit 5: Software Design principles and Methodology (12 Lectures)

Top down and bottom up approach, External Design, Architectural Design and Detailed design, Concept of Cohesion in software design, Classification of Cohesions, Basic concept of Coupling, Classification of Couplings, Introduction to software Analysis and Software Design (SA/SD), Introduction to Data Flow Diagram, Symbols used in DFD, Context Diagram in DFD, Advantages and Disadvantages of DFDs., Balanced DFD, Structured Design: *Transaction Analysis* and *Transform Analysis.* Need of Object Oriented Design and Analysis, UML (Unified Modeling Language), different views of UML, Various UML Diagrams: *Use Case diagram, Class Diagram, Object Diagram*, *Sequence Diagram* and *Collaboration diagram.*

Unit 6: Coding and Testing (9 Lectures)

Goals of coding, Code Review techniques: Code Walkthrough, Code Inspection, Definition of Test cases, test suits, negative testing and positive testing. Different levels of software testing: *unit testing*, *Integration Testing*, *System Testing* and *acceptance testing*. Differentiation between Verification and Validation, Black box testing approaches: *Equivalent Class Partitioning* and *Boundary Value Analysis*, White Box testing approaches: *Statement Coverage*, *Branch Coverage, Condition* Coverage and Path *Coverage. Approach*, MaCabe's Cyclomatic Complexity, Basic idea of various system testing approaches: *Smoke testing, Stress testing, Volume testing* and *Compatibility testing*

Unit 7: Software Reliability and Maintenance (6 Lectures)

What is reliability? Reliability metrics of Software Products: ROCOF, MTTF, MTTR, MTBF, POFOD and availability. ISO 9000 Certification, need of ISO Certification, How to get ISO 9000 certification, Definition of Software Maintenance, Types of Software maintenance: *Corrective, Adaptive* and *Perfective* maintenance, Estimation of Software Maintenance Cost.

COM0500304: Web Technologies

1. Learning Outcome: At the end of the course, students will be able to:

- Understand the basic concept of web applications and web services.
- Design basic well-structured web page using HTML and CSS
- Develop the ability to implement interactive elements and dynamic content using basic JavaScript
- Develop a foundational understanding of server-side scripting using PHP

2. COURSEOUTCOMES:

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

CO1: Define and describe the working principle of the internet and World Wide Web.
CO2: Analyze a web page and identify its elements and attributes.
CO3: Apply HTML and CSS in designing a website and web application.
CO4: Implement server-side functionality using PHP and JavaScript to create interactive web applications.
CO5: Develop a web project and identify its elements and attributes and build customized web sites and web applications.

3. Prerequisites: NIL

4. Semester: 5

5. Course Type: Compulsory

6. Course Level: 300-399

7. Theory Credit: 3

8. Practical Credit: 1

9. No of Hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) Jackson J.C. (2007). *Web Technologies: A Computer Science Perspective*. Pearson.
b) Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
c) Robbins, J. N. (2018). *A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. O'Reilly Media.
d) Robbins, J. N. (2018). *Learning Web Design: A Beginner's Guide.* O'Reilly Media.
e) Haverbeke, M. (2018). *Eloquent JavaScript*. No Starch Press.
f) Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley Professional.

11. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit 1: Introduction to Web Technologies (8 Lectures)

Concepts of the Internet and the World Wide Web (WWW), Overview of web browsers and their functionalities. Client-Server Architecture in Web Applications. Communication Protocols – HTTP, HTTPS, FTP. Working of DNS. Brief concepts of port, URL, cache and cookies. Web Content Accessibility Guidelines. Privacy concerns and data protection regulations, GDPR. Introduction to Web Hosting and control panels.

Unit 2: Front End Development using HTML (10 Lectures)
Website and Webpage. Basic concept of Markup Language. Introduction to HTML. Basic HTML structure. Text formatting Tags – headings, paragraph, line break, horizontal rule. Link and Navigation – anchor tags. Lists - ordered, unordered, definition list. Image and multimedia tags. Tables in HTML. Forms and Input types – text, email, password, radio, select, checkbox, textarea, date, url, submit, button. Semantic HTML. Sectioning elements – header, nav, main, section, article, aside, footer.

Unit 3: Front End Design using CSS (9 Lectures)
Introduction to CSS. CSS syntax and rule structure. Inline, Internal and External CSS. CSS selectors – element, class, ID, attribute. Combinators – descendant, child, adjacent sibling, general sibling. Understanding the CSS Box Model – content, padding, border, margin. CSS colours and backgrounds – background-color, background-image, background-repeat. CSS typography – font properties, text properties.

Unit 4: Client-Side Scripting with JavaScript (10 Lectures)
JavaScript as a high-level interpreted language. JavaScript code execution in web browsers – JavaScript execution context. JavaScript syntax and datatypes. JavaScript variables – var, let, const. Assignment and scope of JavaScript variables. Operators in JavaScript – arithmetic, comparison, logical, assignment. Conditional Statements. Looping Structures. Function declaration and Invocation in JavaScript. Introduction to the Document Object Model. Accessing HTML elements in DOM – by id, by tag name, by class name, query selectors. Manipulating DOM elements – create, add, append, remove. InnerText vs InnerHTML. Manipulating CSS styles using DOM. Event handling and delegation with the DOM using JavaScript. Client-side form validation using JavaScript. Handling form validation and processing data.

Unit 5: Server-Side Programming with PHP (8 Lectures)
Introduction to PHP and role in Web development. PHP syntax and variables. Basic PHP functions – Built-in PHP functions, string manipulation functions, mathematical functions, date and time functions. PHP forms and form handling. Form submission methods – GET and POST. Handling form data with PHP. Uploading files with PHP. Introduction to the tech-stack. Role of Apache, PHP, MySQL etc. Introduction to Databases and SQL. Connecting to databases with PHP. Executing SQL queries with PHP. Retrieving, inserting, updating and deleting data from databases using PHP.

B. List of Practical

(This is a suggestive list only. Questions need not be restricted to this list.)

1. Create a basic HTML webpage structure with a heading, paragraph, and an image.
2. Build a navigation menu using an unordered list (<ul>) with clickable links.

3. Implement a form with input fields for name, email, and a submit button.
4. Create a table with multiple rows and columns to display tabular data.
5. Design an image gallery using HTML and CSS with proper padding and border.
6. Embed a YouTube video on a webpage using the <iframe> tag.
7. Implement an ordered list (<ol>) to display a step-by-step tutorial or instructions.
8. Create a dropdown select menu (<select>) with multiple options.
9. Use HTML5 semantic tags (such as <header>, <nav>, <section>, <article>, <footer>) to structure and organize content on a webpage.
10. Build a registration form with fields for name, email, password, date of birth, address and other such fields with a submit button. Include appropriate input types, labels and placeholders.
11. Style a heading element with a custom font, colour and background.
12. Apply different background colors to alternate rows in a table.
13. Implement a hover effect on a button that changes its background colour or adds a solid border.
14. Style a form input field with custom border, padding, and background color.
15. Implement a CSS tooltip that displays additional information when hovering over an element.
16. Build a simple JavaScript calculator that can perform basic arithmetic operations.
17. Create a button that, when clicked, appends a new paragraph element with a specific text content to an existing div element.
18. Implement a function that changes the innerText of a paragraph element to display a random number between 1 and 10 every time a button is clicked.
19. Build a form with input fields for name and email. When the form is submitted, use innerHTML to display a confirmation message with the entered name and email on the webpage.
20. Build a form with input fields for email, password and confirm password. When the form is submitted, use an alert to display a success message if the password and confirm password values matches, otherwise show an error alert. Use JavaScript for the validation.
21. Create a list of items. Add a click event listener to each item so that when clicked, the background color of the clicked item changes.
22. Write a PHP script to display the current date and time on a webpage.
23. Write a PHP script to connect to a MySQL database and fetch data from a table.
24. Create a registration form with fields for username, email, and password. Implement server-side validation to check for duplicate usernames or invalid email formats. Store the user registration data in a MySQL database. Provide feedback to the user upon successful registration or display appropriate error messages.
25. Design a webpage that displays a list of notices retrieved from a MySQL database. Implement functionality to add new notices to the database using a form. Allow users to view and delete individual notices. Apply appropriate styling to the notices and ensure proper validation and sanitization of user input.

COM0500404: Internship

1. **Learning Outcome:**
    a. Utilize programming, software development, or data analysis skills in a real-world environment.
    b. Identify and troubleshoot technical issues in projects or software systems.
    c. Assess the effectiveness of algorithms, frameworks, or tools used in the industry.
    d. Develop a functional project, application, or report demonstrating internship learnings.
2. **COURSEOUTCOMES:**
   Attendees of the course, students will be able to:

   CO1: Explain industry workflows, best practices, and professional ethics in a computing environment.
   CO2: Implement programming, debugging, or data analysis techniques in real-world projects.
   CO3: Evaluate system performance, identify inefficiencies, and suggest improvements.
   CO4: Develop a project or technical report showcasing applied skills and problem-solving abilities.

3. **Prerequisite:** NIL
4. **Semester: 5**
5. **Course Type: Compulsory**
6. **Course Level: 300-399**
7. **Theory Credit: 0**
8. **Practical Credit: 4**
9. **Number of required hours:**
   a) Theory: 0 hrs
   b) Practical: 60 hrs
   c) Non Contact: 0hrs

10. COURSE CONTENT:

Introduction

Students will have the opportunity to participate in summer internships with local industries, businesses, artists, and craftspeople. Additionally, they can engage in research internships with faculty members and researchers at their institution or other universities and research centers. These internships will allow them to apply their learning in real-world settings, gain hands-on experience, and enhance their employability.

Categories of internship

There will be two categories of internship:

1. **Internship for enhancing employability**:
   Minimizing the gap between theoretical knowledge gained from learning and practical IT skills, enabling graduates to acquire the necessary attributes to join the workforce. This effort may involve collaboration with IT companies, startups, and higher education institutions to develop web applications, mobile applications, cloud service systems, IoT and hardware systems, as well as real-world AI-based tools and blockchain systems.
2. **Internship for developing research aptitude:**
   Providing exposure to actual research environment and develop skills in research tools and techniques including theoretical computer science,

Duration and timing of internship

Credit allocation

COM0600104: Automata Theory and Languages

1. Learning Outcome: After completing this course, students

- Understand the Mathematical model of a finite state machine. Know deterministic and non-deterministic versions of Finite automata.
- Grasp the mathematical concepts of languages and grammar.
- Know Pushdown Automata and the associated grammar/language.
- Know the properties of Regular languages and Context free languages.

2. COURSEOUTCOMES: At the end of the course, students will be able to

CO1: Define the concept of finite automata and language accepted by it.
CO2: Discuss regular grammar and the properties of regular languages.
CO3: Explain the concept of PDA and language accepted by it.
CO4: Analyze the simplification of context free language and Normal forms of context free language.
CO5: Formulate the concept of pumping lemma for regular language and context free language

3. Prerequisites: NIL
4. Semester: 6
5. Course Type: Compulsory
6. Course Level: 300-399
7. Theory Credit: 4
8. Practical Credit: 0
9. No of Hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) *An introduction to Formal Languages and Automata*, Peter Linz,Narosa.
b) *Introduction to Automata Theory, Languages and Computation*, Hopcroft, Motwani and Ullman, Pearson.
c) *Theory of Computer Science (Automata, Languages and Computation*),K. L. P. Mishra, N. Chandrasekaran; P. H.I.

11. Contents of Syllabus:
A. Theory
Detailed Syllabus:
UNIT 1: Finite Automata (10 Lectures)
DFA, NFA, NFA with empty-moves, Equivalence of DFA and NFA, Reduction of the number of states in finite automata.

UNIT 2: Regular Languages and Regular Grammar (12 Lectures)

Concept of languages and grammar, Regular expressions, Connection between regular expressions and regular languages, Regular grammars, Right and Left-Linear Grammars, Equivalence between Regular languages and Regular grammars.

UNIT 3: Properties of Regular Languages (13 Lectures)

Closure under simple set operations- union, intersection, concatenation, complementation and star closure, Decision algorithms for emptiness, finiteness and infiniteness, equality, Proof of non- regularity using Pigeonhole principle and using pumping lemma for regular languages.

UNIT 4: Context Free languages (15 Lectures)

Context-free grammars, leftmost and rightmost derivations, derivation trees, parsing and Ambiguity in grammars and languages, Simplification of Context free Grammars- removing useless productions, empty-productions and unit-productions. Normal forms- Chomsky and Greibach normal forms, Pumping Lemma for CFL, Using Pumping Lemma to show that certain languages are not Context free.

UNIT 5: Pushdown Automata (10 Lectures)

Definition and language accepted (acceptance by empty stack and final state and their equivalence), Pushdown Automata and Context free languages. Deterministic PDA and Deterministic Context free Languages.

COM0600204: Cloud Computing

1. Learning Outcome:

After completing this course, students will know about cloud computing environment, it need and applications.

2. COURSEOUTCOMES: At the end of the course, students will be able to

CO1: Describe client server architecture.

CO2: Illustrate the cloud computing architecture.

CO3: Explain about different Service Level Agreements (SLAs).

CO4: Illustrate data management techniques and security methods used in cloud computing.

CO5: Formulate different case studies on open source and commercial clouds

3. Prerequisites: NIL

4. Semester: 6

5. Course Type: Elective

6. Course Level: 300-399

7. Theory Credit: 4

8. Practical Credit: 0

9. No of Hours:

a) Theory: 60 hrs

b) Practical: 0 hrs

c) Non Contact: 5 hrs

10. List of Books:

a) *Cloud Computing: Principles and Paradigms*, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley,2011

b) *Enterprise Cloud Computing - Technology*, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010

c) *Cloud Computing Bible*, Barrie Sosinsky, Wiley-India, 2010

d) *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*, Ronald L. Krutz, Russell Dean Vines, Wiley- India,2010

e) *Cloud computing,* Ashish Bhatnagar, KATSON Books.

f) *NPTEL :Cloud computing*, By Prof. Soumya Kanti Ghosh, IIT Kharagpur

11. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit 1: Introduction to Cloud Computing (10 Lectures)

Introduction, Definition, basic concepts and terminology, characteristics, goals and benefits, risks and challenges, historical developments, clouds types, Role of networks in cloud computing, Virtualization Technology, Enterprise knowledge clouds, Cloud Computing(NIST Model), Client server Architecture, Client server model vs. Cloud model.

Unit 2: Cloud Computing Architecture (10 Lectures)

Introduction, Cloud Computing stack, Service models(XaaS) : Infrastructure as a Services(IaaS), Platform as a service(PaaS), Software as a Service(SaaS), Application of XaaS, Deployment Models, Microsoft Azure vs Amazon EC2

Unit 3: Service Management in Cloud Computing (10 Lectures)

Service Level Agreements(SLAs), SLA contents, Web Service SLA, Difference between Cloud SLA and Web service SLA, Types of SLA, Service level objectives, Service level management, Considerations for SLA, SLA requirements, Cloud properties: Economic viewpoint

Unit 4: Data Management in Cloud Computing (10 Lectures)

Introduction: Relational database, Google File system, BigTable, MapReduce, Data Storage Techniques, Looking at Data, Scalability & Cloud Services, Database& Data Stores in Cloud, Large scale data processing, Parallel database.

Unit 5: Cloud Security (10 Lectures)

Security – Basic components, Security attacks, Infrastructure Security, Data Security and Storage, Identity and Access Management, Access control, Trust, Reputation, Risk.

Unit 6: Case Study on Open Source and Commercial clouds (10 Lectures)

OpenStack, OpenStack Capability, OpenStack History, OpenStack Architecture, OpenStack components, Meghamala(IITKGP), Google Cloud Platform, Microsoft Azure

COM0600304: Complier Design

1. Learning Outcome:

a) Use compiler construction tools and describes the Functionality of each stage of compilation process
b) Construct Grammars for Natural Languages and find the Syntactical Errors/Semantic errors during the compilations using parsing techniques
c) Analyze different representations of intermediate code.
d) Construct new compiler for new languages.
e) Participate in GATE, PGECET and other competitive examinations

2. COURSEOUTCOMES: At the end of the course, students will be able to

CO1: Describe about compilers, its phases and working of compilers.
CO2: Computer Regular Expressions, NFA, DFA to recognized tokens.
CO3: Design different parsers such as Top-down parser, Bottom-up parsers like LR, SLR and LALR
CO4: Explain intermediate representation i.e. generation of three-address codes, representation of it using DAG and devise code optimization solution for a given problem.
CO5: Develop a Lexical analyzer using Lex and a parser generator using Yacc.

3. Prerequisites: NIL

4. Semester: 6

5. Course Type: Compulsory

6. Course Level: 300-399

7. Theory Credit: 4

8. Practical Credit: 0

9. No of Hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) *Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman* (2007), *Compilers:Principles, Techniques and Tools*, 2nd edition, Pearson Education, New Delhi, India.
b) Alfred V. Aho, Jeffrey D. Ullman (2001), *Principles of compiler design*, Indian student edition, Pearson Education, New Delhi, India.
c) *Kenneth C. Louden (1997),Compiler Construction– Principles and Practice*, 1st edition, PWS Publishing.
d) *K. L. P Mishra, N. Chandrashekaran (2003*), *Theory of computer science- Automata Languages and computation*, 2nd edition, Prentice Hall of India, New Delhi, India.
e) *Andrew W. Appel (2004*), *Modern Compiler Implementation C*, Cambridge University Press, UK.
f) *John R. Levine, Tony Mason, Doug Brown*, *Lex &Yacc*, O’reilly

11. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Introduction to Compiler (12 Lectures)

Definition of compiler, Phases of a compiler, Lexical analysis, Role of lexical analyzer, Tokens, Patterns, Lexemes, Input buffering, Specification of tokens-strings and languages, operations on languages, regular expressions, regular definitions, Recognition of tokens, Lexical analyzer generator- Lex, Finite automata, From Regular expressions to automata.

UNIT 2: Syntax Analysis (16 Lectures)

Parsing, Role of parser, Context free grammar, Parse tree and derivations, Ambiguity, Eliminating ambiguity from dangling-else grammar, Elimination of left recursion, Left factoring,

Top down Parsing- Recursive descent parser, Predictive parser- LL (1) Grammar, construction of predictive parsing table.

Bottom Up Parsing- Reductions, Handle pruning, Shift-Reduce parsing, Conflicts during shift-reduce parsing, LR Parser-Items, Kernel items, Non-kernel items, closure of Item Sets, The function GOTO, LR (0) automaton, Construction of SLR parsing table, Basics of LALR parser, Automatic parser generator-YACC.

UNIT 3: Syntax Directed Translation (12 Lectures)

Syntax directed definition- inherited and synthesized attributes, evaluating an SDD at the nodes of a parse tree, Evaluation orders of SDD's- dependency graphs, ordering the evaluation of attributes, S-attributed and L-attributed definitions, Applications of syntax-directed translation- construction of syntax trees, the structure of a Type, Syntax directed translation schemes- postfix translation schemes, SDT's with actions inside productions, eliminating left recursion from SDT's, Variants of syntax trees- directed acyclic graphs (DAG) for expressions, The value-number method for constructing DAG's, Three address code- Quadruples, Triples and Indirect triples, Static single- assignment form, Types and Declarations, Translation of expressions, Type Checking, Basics of Control flow, Basics of Back patching.

UNIT 4: Run Time Environments (10 Lectures)

Storage organization, Stack allocation of space, Access to non-local data on Stack, Basics of Heap management, Basics of garbage collection

UNIT 5: Code Generation and optimization (10 Lectures)

Machine dependent code generation, Issues in design of code generator, The target language, Addresses in the target code, Basic blocks and flow graphs, Optimization of basic blocks- the DAG representation of Basic blocks, Finding local common sub-expression, dead code elimination, A simple code generator, Basics of Peephole optimization, The Principal Sources of Optimization, Introduction to Data-Flow Analysis.

COM0600404: Artificial Intelligence

1. Learning Outcome:

After completing this course, students will know-

- The fundamentals of artificial intelligence (AI),
- Identify problems where artificial intelligence techniques are applicable.
- Able to apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.

2. COURSEOUTCOMES:

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

CO1: Describe important historical and current trends addressing artificial intelligence.
CO2: Identify and distinguish heuristics searching techniques.
CO3: Apply logic programming concepts in AI.
CO4: Assess the performance and reliability associated with uncertainty and inconsistencies within AI systems.
CO5: Analyze the structure and interpretation of natural languages.

3. Prerequisites: NIL

4. Semester: 6

5. Course Type: Elective

6. Course Level: 300-399

7. Theory Credit: 3

8. Practical Credit: 1

9. No of Hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) *Rich & Knight, Artificial Intelligence* – Tata McGraw Hill, 2nd edition, 1991.
b) *Russell &Norvig, Artificial Intelligence*-A Modern Approach, LPE, PearsonPrentice Hall, 2nd edition, 2005.
c) *W.F. Clocksin and Mellish, Programming in PROLOG,* Narosa Publishing House,3rd edition, 2001.
d) *DAN.W. Patterson, Introduction to A.I and Expert Systems* – PHI, 2007.
e) *Ivan Bratko, Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 3rd edition, 2000.

11. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Introduction (4 Hours)

Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behaviour and environment.

UNIT 2: Problem Solving and Searching Techniques (16 Hours)

Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.

UNIT 3: Knowledge Representation (14 Hours)

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs.
Programming in Logic (PROLOG)

UNIT 4: Dealing with Uncertainty and Inconsistencies (6 Hours)

Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

UNIT 5: Understanding Natural Languages (5 Hours)

Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.

Practical:

Write a prolog program to calculate the sum of two numbers.

Write a prolog program to find the maximum of two numbers.

Write a prolog program to calculate the factorial of a given number.

Write a prolog program to calculate the nth Fibonacci number.

Write a prolog program, insert_nth (item, n, into_list, result) that asserts that result is the list into_list with item inserted as the nth element into every list at all levels.

Write a Prolog program to remove the nth item from a list.

Write a Prolog program, remove nth (Before, After) that asserts the After list is the Before list with the removal of every nth item from every list at all levels.

Write a Prolog program to implement append for two lists.

Write a Prolog program to implement palindrome (List).

Write a Prolog program to implement max(X,Y,Max) so that Max is thegreater of two numbers X and Y.

Write a Prolog program to implement maxlist(List,Max) so that Max is thegreatest number in the list of numbers List.

Write a Prolog program to implement sumlist(List,Sum) so that Sum is thesum of a given list of numbers List.

Write a Prolog program to implement two predicates evenlength(List) and oddlength (List) so that they are true if their argument is a list of even or oddlength respectively.

Write a Prolog program to implement reverse (List, Reversed List) thatreverses lists.

Write a Prolog program to implement maxlist (List, Max) so that Max is the greatest number in the list of numbers List using cut predicate.

Write a Prolog program to implement GCD of two numbers.

Write a prolog program that implements Semantic Networks/Frame Structures.

COM0600504: Computer Graphics

1. Learning Outcome:

After completing this course, students will know about-

- Basic elements of Computer Graphics
- Fundamental of Computer graphics algorithms along with basic mathematical foundations of computer graphics.

2. COURSEOUTCOMES:

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

CO1: Describe basic knowledge of core concepts of computer graphics input/output systems.
CO2: Explain different techniques for designing computer graphics software.
CO3: Design and construct complex geometric models using advanced modeling techniques.
CO4: Utilize visible surface detection algorithms to render realistic scenes.
CO5: Apply basic illumination models to simulate lighting effects and enhance the visual realism of rendered scenes.

3. Prerequisites: NIL
4. Semester: 6
5. Course Type: Elective
6. Course Level: 300-399
7. Theory Credit: 3
8. Practical Credit: 1
9. No of Hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

10. List of Books:

a) D. Hearn, M. Baker: Computer Graphics, Prentice Hall of India 2008.
b) J.D.Foley, A.Van Dan, Feiner, Hughes Computer Graphics Principles & Practice2nd edition Publication Addison Wesley 1990.
c) D.F.Rogers Procedural Elements for Computer Graphics, McGraw Hill 1997.
d) D.F.Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill, 2nd edition 1989.

11. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Introduction **(2 Hours)**

Basic elements of Computer Graphics, Applications of Computer Graphics

UNIT 2: Graphics Hardware (5 Hours)

Input Devices: Keyboard, Mouse, Trackball & Space ball, Joystick, Data Glove, Digitizers, Image Scanners, Touch panels, Light Pens systems. Output display devices: Refresh CRT, Raster-Scan display and Random-scan display technique, Color display techniques-Beam penetration method and Shadow-mask method, Direct view storage tubes, Emissive & Non-emissive flat-panel, Displays-Plasma panels, LED and LCD monitor, Three-dimensional viewing devices and Virtual-Reality systems Display processor: Raster-scan systems, Random-scan systems

UNIT 3: Fundamental Techniques in Graphics (20 Hours)

Line-drawing algorithms: DDA algorithm and Bresenham's Line drawing Algorithm, Midpoint Algorithm for Circle and Ellipse Generation, Curve generation. Attributes for output primitives: Area-filling Algorithms - Scan-line Polygon-fill, 2-D Geometric Transformations: Basic transformations-translation, Rotation and Scaling Matrix representations and Homogeneous Co-ordinate representations, Composite transformations among translation, Rotation and Scaling, 2-D viewing: Definition, Viewing transformation pipeline, Window-to-viewport Co-ordinate transformation.

2-D Clipping: Concept and Algorithm: Point clipping, Line clipping - Cohen-Sutherland algorithm, Area clipping, Text clipping, Polygon clipping.3-D concepts: Display methods-Parallel projection, perspective projection3-D geometric transformations: Transformation, Translation, Rotation and Scaling around axes, 3-D Viewing Projections – Parallel and Perspective.

UNIT 4: Geometric Modelling (8Hours)

Representing curves and surface, Bezier curves and surfaces – Definition of Bezier curve and its properties, Algorithms for Bezier curves and surfaces, Hermite curve

UNIT 5: Visible Surface determination (5Hours)

Definition, approaches for visible surface detection, object-space methods- Back-Face Detection,

Image space methods: Depth Buffer Methods, A Buffer Method, Scan Line Method, Depth-Sorting Method

UNIT 6: Surface rendering (5Hours)

Definition and importance, light sources, Basic illumination models-Ambient light, Diffuse reflection, Specula reflector and Phong model

Practical:

- Write a program to implement DDA algorithm for line drawing.
- Write a program to implement Bresenham's line drawing algorithm.
- Write a program to implement mid-point circle drawing algorithm.
- Write a program to clip a line using Cohen-Sutherland line clipping algorithm.
- Write a program to clip a polygon using Sutherland Hodgeman algorithm.
- Write a program to apply 2D translation on a 2D object (use homogenous coordinates).
- Write a program to apply 2D rotation on a 2D object (use homogenous coordinates).

- Write a program to apply 2D scaling on a 2D object (use homogenous coordinates).
- Write a program to apply 2D reflection of a 2D object (use homogenous coordinates).
- Write a program to apply 2D shear operation on a 2D object (use homogenous coordinates).
- Write a program to apply 3D translation on a 3D object (use homogenous coordinates).
- Write a program to apply 3D rotation on a 3D object (use homogenous coordinates).
- Write a program to apply 3D scaling on a 3D object (use homogenous coordinates).
- Write a program to apply 3D reflection of a 3D object (use homogenous coordinates).
- Write a program to apply 3D shear operation on a 3D object (use homogenous coordinates).
- Write a program to draw Hermite/Bezier curve.

COM0600604: Data Mining and Warehousing

1. Learning Outcome:

a) Understanding the process of Knowledge Discovery in Databases
b) Understand the functionality of the various data warehousing component.
c) Characterize the kinds of patterns that can be discovered by association rule mining.
d) Analysis of different types of data by clustering and classification.

2. COURSE OUTCOMES:

CO1: Describe the components and architecture of data warehouse architecture
CO2: Illustrate different data mining techniques such as association rule mining, clustering and classification.
CO3: Develop various algorithms like A priori, Pincer Search and DIC used in association rule mining
CO4: Explain the working principle of various clustering and classification algorithms like K-means, K-Mediods, PAM, DBSCAN, K-Nearest Neighbor Classifiers etc.
CO5: Analyze the uses of web mining, text mining, big data mining, neural networks etc.

3. Prerequisites: NIL
4. Semester: 6
5. Course Type: Elective
6. Course Level: 300-399
7. Theory Credit: 3
8. Practical Credit: 1
9. No of Hours:

a) Theory: 45 hrs
b) Practical: 30 hrs
c) Non Contact: 5 hrs

9. List of Books:

a) A.K. Puzari, *Data Mining Techniques*, University Press.
b) J. Han, J. Pie and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann.
c) P. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, Pearson Education (LPE).
d) G. K. Gupta, *Introduction to Data Mining with Case Studies*, PHI.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Overview (4 Lectures)

What is Data Mining?, Knowledge Discovery in Databases (KDD) vs. Data Mining, Types of Data, Basic Data Mining Tasks, Predictive and Descriptive data mining techniques, Supervised and Unsupervised learning techniques, Basics of Pre-processing methods- Data Cleaning, Data Integration and Transformation, Data Reduction, Data Visualization.

UNIT 2: Data Warehousing (6 Lectures)

What is Data Warehouse? Multidimensional Data Model, Data Cube, Basic Components of Multidimensional Data Model, OLAP Operations- Slicing, Dicing, Drilling, Drill-Up, Drill-Down, Drill-Within, Drill-Across, Pivot(Rotate), Schema of Warehouse, Data Warehouse Architecture, Metadata.

UNIT 3: Association Rule Mining (12 Lectures)

What is Market Basket Data?, k-Itemset, Support of an Itemset, Frequent Itemsets, Infrequent Itemsets, Maximal Frequent Itemsets, Closed Frequent Itemsets, Association Rules, Confidence of a Rule, Problem of Mining Association Rules, Algorithm for Mining Frequent Itemsets- Apriori Algorithm, Pincer-Search Algorithm, DIC (Dynamic Itemset Counting) Algorithm, Steps of Mining Association Rules.

UNIT 4: Clustering (12 Lectures)

What is Clustering, Partitional vs Hierarchical Clustering, Types of Data in Clustering, Distance Measures used in Clustering- Euclidean Distance, Manhattan Distance, Similarity Measures used in Clustering- Cosine Similarity, Jacquard Coefficient, Partitional Clustering Methods- K-Means, K-Mediods, PAM, CLARA, CLARANS, Density Based Clustering Methods- DBSCAN, Introduction to Hierarchical Clustering.

UNIT 5: Classification (8 Lectures)

What is Classification? Issues Regarding Classification, K-Nearest Neighbor Classifiers, Bayesian classification, Introduction to Decision Tree.

UNIT 6: Recent Trends and Techniques used in Data mining (3 Lectures)

Basic Concepts of- Web Mining, Spatial Data Mining, Temporal Data Mining, Big Data Mining, Concept of Neural Network, Genetic Algorithm.

Practical / Lab work to be performed

- Implement ***any one*** from the following-
    - Write a computer program to implement A priori algorithm to mine all frequent itemsets from a transactional dataset. Use hashing to store the item sets in the level wise generation of candidate sets.
    - Write a computer program to implement the Pincer Search algorithm.
    - Write a computer program to implement the DIC (Dynamic Item set) algorithm.
- Implement ***any four*** from the following-
    - Write computer program to implement the K-Means algorithm using different distance measures stated in the syllabus.
    - Write computer program to implement the PAM algorithm using different similarity measures stated in the syllabus.
    - Write a computer program to implement the CLARA algorithm.
    - Write a computer program to implement the CLARANS algorithm.
    - Write a computer program to implement the DBSCAN algorithm.
    - Write a computer program to implement the K-NN algorithm.

COM0600704: PROJECT AND PRESENTATION

11. Learning Outcome:

- Students will recall and describe the problem statement, objectives, and methodologies employed in their project.
- Students will demonstrate an understanding of the technologies explored during the project, explaining their relevance, functionality, and potential applications.
- Students will apply the acquired knowledge and skills to develop a solution or prototype addressing the identified problem, utilizing the chosen technologies effectively.
- Students will critically analyze the project outcomes, assessing the strengths and weaknesses of their approach, and identifying areas for improvement or further exploration.
- Students will synthesize their findings into a coherent dissertation, presenting their research methodology, results, and conclusions while evaluating the implications and significance of their work within the broader context of the field.

12. COURSE OUTCOMES:

Attendees of the course, students will be able to:

CO1: Students will be able to recall and recognize fundamental concepts, principles, and terminology relevant to the course.
CO2: Students will demonstrate a comprehensive understanding of theoretical frameworks, methodologies, and models discussed in the course.
CO3: Students will be able to apply acquired knowledge and skills to solve practical problems, analyze case studies, and develop solutions in real-world scenarios relevant to the course content.
CO4: Students will demonstrate the ability to synthesize diverse sources of information, integrate multiple perspectives, and propose innovative solutions or designs.
CO5: Write their findings and analysis in the form of a dissertation.

13. **Prerequisite:** Basic Subject knowledge
14. Semester: 6
15. Course Type: Compulsory
16. Course Level: 300-399
17. Theory Credit: 0
18. Practical Credit: 4
19. Number of required hours:

a) Theory: 0 hrs
b) Practical: 60 hrs
c) Non Contact: 0hrs

20. COURSE CONTENT:

At the onset of their sixth semester, each student will receive an assignment for a project. Students, either individually or in pairs, will delve into a unique problem under the mentorship of a faculty member from the department. The chosen problem should allow students to delve deeply into one or two specific technologies, fostering a strong understanding and proficiency in those areas upon project completion.

To promote innovation and avoid redundancy, previously tackled problems should be avoided unless they hold exceptional research significance and expansive scope. While application-based problems spurred by specific demands may be considered, simplistic information management systems comprising only a few database tables or data entry forms should be discouraged.

Interdisciplinary collaboration where applicable, enabling students to draw insights from diverse fields and perspectives to enrich their projects will be encouraged. Students also have the option to conduct their projects in collaboration with other institutes or organizations, subject to approval from the relevant institute organization. However, at least one project supervisor must be affiliated with the institute or organization.

Regular progress updates must be reported by meetings with the project supervisor throughout the project duration.

Students should look for opportunities to publish their project findings in academic journals, conferences, or other relevant platforms to disseminate their research outcomes and contribute to the academic community.

Projects must culminate in the submission of a dissertation. Evaluation and presentation of projects will adhere to the regulations outlined in the PG course semester system of G.U., with choice-based credit and grading system.

21. COURSE ASSESSMENT DETAILS:

Internal assessment: seminars, presentations, viva, project implementation

COM0700104: Research Methodology

1. **Learning Outcome:**
   - To introduce the basic concepts in research methodology in Computer Science.
   - To familiarize the issues inherent in selecting a research problem
   - To discuss the techniques and tools to be employed in completing a research project.
   - Enable the students to prepare report writing and framing Research proposals.

2. **COURSEOUTCOMES:**

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

CO1: Explain the basics in scientific research methodology and its various types

CO2: Apply different data collection and sampling designs, and also to perform statistical analysis of the collected data.

CO3: Develop skills in qualitative and quantitative data analysis and presentation.

CO4: Analyze different research articles/papers for writing critical reviews

CO5: Use ethical approaches in research to obtain maximum benefit and minimum risk to human kind in the research process

3. Prerequisites: NIL

4. Semester:7

5. Course Type: Compulsory

6. Course Level: 400-499

7. Theory Credit: 3

8. Practical Credit: 1

9. No of Hours:

a) Theory: 60 hrs

b) Practical: 0 hrs

c) Non Contact: 5 hrs

10. List of Books

a) C. R. Kothari; (2004); Research Methodology-Methods and Techniques; New-age International Publishers

b) K. Prathapan; (2019); Research Methodology for Scientific Research; Wiley Publication

11. Contents of the Syllabus

UNIT-1 Introduction to Computer Science Research (4 hours)

What is Research? Types of Research, Why Research, Significance & Status of Research in Computer Science.

UNIT-2 Steps in Research (6 hours)

Major Journals & Publications in Computer Science, Major Research Areas of Computer Science, Identification, selection & formulation of research problems. Developing a research proposal, planning your research. How engineering and technological research differs from other scientific research.

UNIT-3 Research Data (6 hours)

Data Collection: Methods of Data Collection, Theory of Sampling, Sampling techniques. Size of a sample.

UNIT-4 Data Analysis and interpretation (12 hours)

Statistical analysis of data: Measures of central tendency, dispersion; Associations/Relations; Regression and Co-relation analysis; Hypothesis testing and tests of significance. Data processing software and statistical inference, Interpretation of results.Use of R-Programming in Statistical Analysis and Data Visualization.

UNIT-5 Simulation and tools (6 hours)

Concept of Simulation. Time and randomness in simulation, Application of simulations. How a simulation model works, tools for simulations.

UNIT-6 Literature Survey (6 hours)

Finding out about your research area, Literature search strategy. Writing critical reviews. Identifying venues for publishing your research.

UNIT-7 Plagiarism (6 hours)

Concept and Importance of understanding what is plagiarism and what is not plagiarism. Methods and ways to detect and avoid plagiarism. Available tools and software for plagiarism check.

UNIT-8 Writing Papers and the Review Process (10 hours)

Preparing and presenting your paper. The conference review process. Making use of the referees' reports. The journal review process, Group exercise in reviewing research papers. Tools for paper formatting and Reference Management. Use of LaTeX.

UNIT-9 Ethical Issues and Professional Conducts (4 hours)

Ethics in general, Professional Ethics, Ethical Issues that Arise from Computer Technology, General Moral Imperatives, More Specific Professional Responsibilities, Organizational Leadership Imperatives.

Practical Sessions:

Practical sessions will be held for the following

(i) R-Programming in Statistical Analysis and Data Visualization.

(ii) Explore available tools and software for plagiarism check.

(iii) Use of LaTeX in writing documents with equations, figures etc. and report generation.

COM0700204: FUNDAMENTALS OF MACHINE LEARNING

1**. Learning Outcomes:** At the end of the course, students will be able to:

(a) Understand what is machine learning and themajor machine learning approaches
(b) Differentiate between supervised and unsupervised learning
(c) Understand linear and non- linear classification.
(d) Understand the role and importance of feature extraction.
(e) Apply machine learning algorithms to real world problems.

2**. Prerequisites:** FundamentalsofPythonprogramming

3**. Semester:** 7

4. **Course type:** Compulsory

5. **Course level:** 400-499

6. **Theory credit:** 3

7. **Practical credit:** 1

8. **Number of required hours**:

a) Theory:35 hrs (35 classes)

b) Practical: 30 hrs (15 classes)

c) Non- Contact: 10

9**. Reference books:**

1. J. Shavlik and T. Dietterich (Ed), Readings in Machine Learning, Morgan Kaufmann, 1990.
2. Chris Albon, Machine Learning with Python Cookbook, Pub: Shroff/O'Reilly, 2018.
3. Shai Shalev-Shwartz and Shai Ben-David; Understanding Machine Learning: From Theory to Algorithms. E-book; https:// kdnuggets.com
4. NPTEL

10. **Course Outcomes**

CO-01	Understand supervised and unsupervised learning
CO-02	Develop programs for implementing and testing algorithms in ML
CO-03	Understand the role of features in classification and the term dimension reduction
CO-04	Understand unsupervised learning using autoencoder
CO-05	Apply ML algorithms to diverse real-world problems.

UNIT I: Introduction to Machine Learning (12 Lectures)

Definition and scope of machine learning, History and Goals of ML. Types of machine learning: supervised, unsupervised, reinforcement learning, Applications of machine learning, Data Preprocessing, Dealing with heterogeneous data. Introduction to Python libraries for machine learning (NumPy, pandas, scikit-learn).

UNIT II: Linear Regression (05 Lectures)

Simple linear regression, Multiple linear regression, Gradient descent, Regularization techniques: L1 and L2 regularization

UNITIII: Supervised Learning and Classification (07 Lectures)

Logistic regression, Decision trees – Entropy, Splitting attribute and Splitting criteria. Support Vector Machines (SVM), k-nearest Neighbors (k-NN) algorithm.

UNIT IV: Model Evaluation and Selection (04 Lectures)

Training and testing datasets, Cross-validation, Evaluation metrics: accuracy, precision, recall, F1-score, ROC curve, AUC.Concept of Ensemble and Ensemble techniques- Bagging, Boosting, Stacking and Voting.

UNIT V: Unsupervised Learning (08 Lectures)

Partitional Clustering - K-means and k-medoids algorithm. Hierarchical clustering – Agglomerative and Divisive. Density based clustering – DBScan algorithm.

UNIT VI: Dimensionality Reduction (06 Lectures)

Feature selection, Feature extraction, Principal Component Analysis (PCA).

UNIT VII (03 Lectures)

Working on real-world datasets to show applications of machine learning in Natural Language Processing (NLP), Image processing, Recommender Systems, Healthcare etc.

COM0700304: ADVANCED OPERATING SYSTEM

1. **Learning Outcomes**: After successful completion of this course, students will be able to:
    - Understand advanced operating system concepts and architectures.
    - Explore virtualization technologies and their applications.
    - Study real-time operating systems and their characteristics.

2. **COURSE OUTCOMES**:

   At the end of the course, students will be able to:
   CO1: Identify different operating advanced systems and its types, functions, applications.
   CO2: Explain process, threads and solve process scheduling problems of various advanced operating systems.
   CO3: Describe hardware requirements, load balancing of advanced operating systems.
   CO4: Compile case study reports of various real world advanced operating systems.
   CO5: Compare advanced operating systems and utilize them according to their use cases.
3. **Prerequisites: Operating Systems**
4. **Semester: 7**
5. **Course Type: Elective**
6. **Course Level: 400-499**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **No of required hours:**

   Theory: 60 hours
   Practical: 0 hours
   Non Contact: 5 hours

10. **List of Reference Books:**
    - Modern Operating Systems, Andrew S. Tanenbaum, 4th edition, Pearson Education
    - Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne,Wiley
    - Distributed Systems Principles and Paradigms Tanenbaum, Steen 2nd edition, Pearson Education
    - Real-Time Systems, Jane W. S. Liu, 1st edition, Pearson Education India

11. **Contents of Syllabus**:

Unit I: Review of Operating system concepts (2 hours)
Process and threads: Process Control Block (PCB), process scheduling, Process synchronization: critical section problem, semaphore, process deadlock.

Unit II: Multiprocessor systems (12 hours)
Introduction to multiprocessor systems, multiprocessor hardware: UMA (Uniform Memory Access), NUMA (Non Uniform Memory Access), Multicore Chips,Moore's law, Manycore chips. Multiprocessor Operating System Types: private OS, Master-Slave Multiprocessors,

Symmetric Multiprocessors, Multiprocessor Synchronization: TSL (Test and Set Lock), Peterson’s protocol, Multiprocessor scheduling: Space Sharing, Gang Scheduling.

Unit III: Multicomputer systems (13 hours)
Introduction to multicomputer systems, multicomputer Hardware organization: interconnection Technology, Network Interfaces, Low-Level Communication Software, User-Level Communication Software: Send and Receive, Blocking versus Nonblocking Calls, Remote Procedure Call : client stub, server stub, marshalling, Load Balancing, processor allocation algorithms: Graph-Theoretic Deterministic Algorithm, Sender-Initiated Distributed Heuristic Algorithm, Receiver-Initiated Distributed Heuristic Algorithm.

Unit IV: Distributed system(10 hours)
Definition of a distributed system. Characteristics of distributed systems, distributed systems vs centralized systems. Examples of Distributed Systems in Real-world Applications, System models: Fundamental and Architectural model, System architectures- client-server architectures, Synchronization: Needs of clock synchronization, external and internal clock synchronization, Logical and vector clocks, Lamport’s logical clock, Vector clocks, Causal Order of messages. Global state, Chandy Lamport snapshot algorithm

Unit V: Virtualization and Containerization (10 hours)
Definition of virtualization, need of virtualization, examples of virtual machines, advantage and disadvantage of virtualization, hypervisor architecture : type 1 and type 2 hypervisor, guest OS, host OS, Introduction to Containerization Technologies (e.g., Docker, Kubernetes), Containers vs. Virtual Machines, benefits of containerization, use cases of containerization, container orchestration

Unit VI: Real-time operating systems (10 hours)
Definition and Characteristics of Real-Time Systems(RTOS), Differences Between General-Purpose Operating Systems and RTOS, Importance of Timing Constraints in RTOS, Types of RTOS, Hard vs Soft RTOS, Classification of Real-Time Tasks: Periodic, Aperiodic, Sporadic tasks, Real-Time Scheduling Algorithms: Earliest Deadline First, Least Laxity First.

Unit VI: Case study on real-time operating systems (3 hours)
FreeRTOS, FreeRTOS features, Applications of FreeRTOS, FreeRTOS architecture, FreeRTOS supported architectures, FreeRTOS libraries.

COM0700404: ADVANCED COMPUTER ORGANIZATION AND ARCHITECTURE

1. **Learning Outcomes:**
   - Understand the importance of multiprocessor and multicomputer.
   - Learn about advance architecture designs.
   - Understand the concepts of interconnected structures.
   - Learn about data flow computer architectures.
2. **Course Outcomes:**

At the end of the course, students will be able to:
CO1: Explain the working of computer arithmetic.
CO2: Demonstrate concepts of parallelism in hardware/software
CO3: Apply assembly language to create instruction sets.
CO4: Describe architectural features of advanced processors.
CO5: Explain data flow in arithmetic algorithms.

3. Prerequisites: Fundamental knowledge of computer Organization and Architecture
4. Semester: 7
5. Course Type: Elective
6. Course Level: 400-499
7. Theory Credit: 4
8. Practical Credit: 0
9. No of Hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non-Contact: 5 hrs

10. List of Books:

- Govindarajalu, B. Computer Architecture and Organization, TMH publication.
- Richard Y. Kain, Advanced Computer Architecture A systems Design Approach, PHI Publication
- Stallings William, Computer Organization and Architecture Designing for Performance, Pearson Education
- Kai Hwang, “Advanced Computer architecture Parallelism, scalability, Programmability”, McGraw-Hill, N.Y, 2003
- Kai Hwang and Faye Briggs, “Computer Architecture and Parallel Processing”, McGraw-Hill International Edition, 2000

11. Contents of Syllabus:

UNIT 1: Computer Arithmetic 10

Serial adder, Parallel adder, Ripple carry adder, Carry look-ahead adder, Multiplication of signed and unsigned numbers, Booth's algorithm, Division of integer, Floating point arithmetic.

UNIT 2: Advanced Architectures 15

Classification: SISD, SIMD, MISD, MIMD, Bus data path:one, two and three bus, Scalar, vector, superscalar and pipelined processor,VLIW architecture, EPIC architecture, Pipelining: Instruction pipeline, pipeline bubbles, Hazards: resource conflicts, data dependency, branch difficulty, Vector computing: arithmetic pipeline, vector and scalar register, chaining, scatter gather operations, vector-register processor, Memory vector processor, Array processor.

UNIT 3: Assembly Language 10

Machine Language, Register transfer language, Assembly Language, Assembler, Program Loops, Subroutines, Developing counters and Time delay routines, Input-Output Programming. Interfacing concepts.

UNIT 4: 8085 Microprocessor Architecture and Memory Interface 10

Pin Description, Operating Modes, Instruction Set of 8085: Instructionset, Data formats, Addressing modes, Opcode & Operands, Data and Storage, Word size, 8085 Instructions: Counter and Time delays, Stack, Subroutines, Call &Return statements, Interrupts in 8085: generation of RST codes, interrupt priority, SIM & RIM instructions.

UNIT 5: Multiprocessor 8

Characteristics of Multiprocessors, Interconnection Structures, Interprocessor Arbitration, Interprocessor Communication and Synchronization, Cache Coherence, Multicore Processors

UNIT 6: Dataflow Computers and VLSI Computations 7

Data flow architecture: static and dynamic, VLSI Computing Structures, Array Architectures, Mapping algorithms, Reconfigurable processor array, VLSI matrix arithmetic models and processors, Matrix Algorithms and Pipelines

COM0700504: CRYPTOGRAPHY AND NETWORK SECURITY

1. **Learning Outcome:**
   - Demonstrate a comprehensive understanding of fundamental concepts, principles, and technologies related to network security, including encryption techniques, authentication protocols, and security architectures.
   - Apply their knowledge and skills to analyze, design, and implement effective security measures to protect networked systems and data from unauthorized access, malicious attacks, and other security threats.

2. **Course Outcome:**
   Atthe end of this course, students will be able to:

   - Students will recall and summarize fundamental concepts of network security, including encryption techniques, authentication protocols, and security threats.
   - Students will demonstrate an understanding of various encryption algorithms and protocols used in network security, explaining their purposes, mechanisms, and limitations.
   - Students will comprehend the importance of key management, digital signatures, and authentication mechanisms in ensuring the confidentiality, integrity, and availability of networked systems.
   - Students will comprehend the importance of key management, digital signatures, and authentication mechanisms in ensuring the confidentiality, integrity, and availability of networked systems.
   - Students will evaluate the strengths and weaknesses of different approaches to network security, assessing their suitability for addressing specific security requirements and mitigating emerging threats.

3. **Prerequisite: NIL**
4. **Semester: 9**
5. **Course Type: Elective**
6. **Course Level: 400-499**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **Number of required hours:**

   a) Theory: 60 hrs

   b) Practical: 0 hrs

   c) Non Contact: 5 hrs

10. **List of reference books:**
    a) W. Stallings, *Cryptography & Network Security Principles & Practices* (7th Edition), Pearson Education, 2017.68
    b) Wade Trappe, Lawrence C Washington, "*Introduction to Cryptography with coding theory*", Pearson.

c) D. R. Stinson, *Cryptography: Theory & Practice (3rdEdition)*, CRC Press, 2006.

11. Detailed Syllabus:

UNIT I: INTRODUCTION 8 Hours

OSI Security Architecture -Security Attacks-Security Services-A Model for Network Security- Classical Encryption techniques – Cipher Principles – Symmetric Cipher Model; Substitution Techniques; Transposition Techniques; Steganography

UNIT II: Modern Block Ciphers 7 Hours

Data Encryption Standard – Block Cipher Design Principles and Modes of Operation - Evaluation criteria for AES – AES Cipher – Triple DES – Placement of Encryption Function – Traffic Confidentiality

UNIT II: PUBLIC KEY CRYPTOGRAPHY AND HASH FUNCTIONS 15 Hours

Public Key Cryptosystems, Applications, Requirements, Cryptanalysis, RSA Algorithm,Key Management - Diffie-Hellman key Exchange – Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Simple Hash Functions, Authentication requirements – Authentication functions – Message Authentication Codes-Message Authentication Requirements, Authentication Functions, Requirements of Message authentication codes, Security of MACs.

UNIT IV: NETWORK SECURITY 15 Hours

Digital Signatures; Key Management and Distribution;Authentication Applications: Kerberos – X.509 Authentication Service – Electronic Mail Security – PGP – S/MIME - IP Security – Web Security; Intrusion detection – password management – Viruses and related Threats – Virus

Counter measures – Firewall Design Principles – Trusted Systems.

COM0700604: Advanced Database Management Systems

1. **Learning Outcome:**
   - Students will be able to recall key concepts and terminologies related to SQL, object databases, XML, distributed databases, and NoSQL systems.
   - Students will analyze the effectiveness and efficiency of distributed database architectures, query processing, and transaction management, as well as the advantages and limitations of different NoSQL systems.
   - Students will evaluate advanced database models and systems, such as active, temporal, spatial, and multimedia databases, and assess data mining techniques for extracting meaningful insights from large datasets.

2. **COURSE OUTCOMES:**

CO1: Recall and comprehend fundamental and advanced SQL concepts, including data definition, constraints, and complex queries.

CO2: Understand object-oriented, object-relational, and XML database concepts and technologies.

CO3: Analyze and evaluate distributed database systems and NoSQL technologies.

CO4: Create and assess advanced database models and systems, including active, temporal, spatial, multimedia, and deductive databases.

3. **Prerequisite:** DBMS - COM040204
4. **Semester:** 7
5. **Course Type:** Elective
6. **Course Level:** 400-499
7. **Theory Credit**: 4
8. **Practical Credit**: 0
9. **Number of required hours**:
   (a) Theory: 60 hrs
   (b) Practical: 0 hrs
   (c) Non Contact: 5 hrs
10. **List of reference books:**
    (a) Elmasri Ramez and Navathe Shamkant; *Fundamentals of Database System*, 7edition, Pearson
    (b) R. Ramakrishnan, J. Gehrke, Database Management Systems, McGraw Hill, 2004

11. Detailed Syllabus:

Unit 1: SQL Revisited **9 Hours**

SQL Data Definition and Data Types; Specifying Constraints in SQL; Basic Retrieval Queries in SQL; INSERT, DELETE, and UPDATE Statements in SQL; Additional Features of SQL; More Complex SQL Retrieval Queries; Specifying Constraints as Assertions and Actions as Triggers; Views (Virtual Tables) in SQL; Schema Change Statements in SQL

Unit 2: Object Database Concepts and XML 12 Hours

Overview of Object Database Concepts- Introduction, Object Identity, and Objects versus Literals, Object Identity, and Objects versus Literals, Encapsulation, Type Hierarchies and Inheritance ; Object Database Extensions to SQL; The ODMG Object Model and the Object Definition Language ODL; Object Database Conceptual Design; The Object Query Language OQL; Overview of the C++ Language Binding in the ODMG Standard; Structured, Semi-structured, and Unstructured Data; XML Hierarchical (Tree) Data Model; XML Documents, DTD, and XML Schema; Storing and Extracting XML Documents from Databases; XML Languages; Extracting XML Documents from Relational Databases; XML/SQL: SQL Functions for Creating XML Data.

Unit 3: Distributed Databases 6 Hours

Distributed Database Concepts; Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design; Overview of Concurrency Control and Recovery in Distributed Databases; Overview of Transaction Management in Distributed Databases; Query Processing and Optimization in Distributed Databases; Types of Distributed Database Systems; Distributed Database Architectures; Distributed Catalog Management;

Unit 4: NOSQL Databases 6 Hours

Introduction to NOSQL Systems; The CAP Theorem; Document-Based NOSQL Systems and MongoDB; NOSQL Key-Value Stores; Column-Based or Wide Column NOSQL Systems; NOSQL Graph Databases and Neo4j

Unit 5: Advanced Database Models and Data mining 12 Hours

Active Database Concepts and Triggers; Temporal Database Concepts; Spatial Database Concepts; Multimedia Database Concepts; Introduction to Deductive Databases; Active Database Concepts and Triggers; Temporal Database Concepts; Spatial Database Concepts; Multimedia Database Concepts; Introduction to Deductive Databases; Trends in Information Retrieval; Overview of Data Mining Technology; Association Rules -Market-Basket Model, Support, and Confidence, Apriori Algorithm, ; Classification; Clustering; Applications of Data Mining;

COM0800104: ADVANCED DATA STRUCTURE

1. **Learning Outcomes:** At the end of the course, students will be able to:

   (a) Understand and apply the fundamental data structures and algorithms – such as arrays, linked lists, stacks, queues, trees, sorting and searching algorithms using C programming language.

   (b) Analyze the time and space complexity of different algorithms and choose the appropriate algorithm for a given problem.

   (c) Develop efficient algorithms to solve various computational problems by utilizing data structures and algorithms covered in the course.

2**. Prerequisites:** Fundamentals of C/C++programming

3. **Semester:** 7

4. **Course type: Compulsory**

5. **Course level: 500-599**

6. **Theory credit: 3**

7. **Practical credit: 1**

8**. Number of required hours:**

   a) Theory:35 hrs (35 classes)

   b) Practical: 30 hrs (15 classes)

   c) Non Contact: 10

9. **Reference books:**

   a) Cormen T.H., Leiserson C.E., RivestR.L.; *Introduction to Algorithms*; Tata-McGraw Hill Publishers
   b) Aho A., HopcroftJ.E.,UllmanJ.D.;*Data Structures and Algorithms*;Addison-Wesley
   c) Horowitz, Sahani; *Fundamentals of Data StructuresinC/C++*;ComputerSciencePress
   d) AhoA.,HopcroftJ.E.,UllmanJ.D.;*Design and Analysis of Computer Algorithms*; Addison-Wesley.
   e) NPTEL

10. **Course Outcomes**

CO-01 Analyzelinearandnon-lineardatastructureslikestacks,queues,linkedlistetc.

CO-02 Solveproblemsrelatedtodatadictionarydatastructuressuchassearchtree,AVLtree,RedBlacktrees,Splay trees and Hashing.

CO-03 ComparedifferentSortingandSearchingtechniquessuchasQuicksort,HeapSort,RadixSort,CountingSort, BST, Median and Order Statistics and Heap data structures.

CO-04 To solve problems of sorting and searching.

CO-05 To extract the time complexity of different algorithms like order statistics, amortized analysis

11**. Detailed Syllabus:**

A. Theory

UNIT-I: Review of basic concepts in Data Structure
A quick review of array versus linked list structure; binary tree, binary search tree; traversal, insertion and deletion in binary search trees.

UNIT-II: **Dictionary ADT**

Search trees, balancing of search trees – AVL trees, Red-Black trees, multi way search trees, 2-3 trees, splay trees. Insertion and Deletion in each of the above data structures. Hashing.

UNIT-III: **Sorting and Selection Techniques**

Quick sort, Heap sort, Shell sort, sorting in linear time – Counting sort, Radix sort. Medians and order Statistics. Selection and Adversary arguments. Lower bound onsorting.

UNIT-IV**: Priority Queue ADT**
Heaps-extended priority queue, min(max) heaps, binomial heap, fibonacci heap and its amortized analysis.

UNIT-V:**PartitionADT**

Union-find algorithms through weighted merge and path compression.

UNIT-VI: **Data Structure for external storage operations**

B-tree, insertion and deletion in B-trees, external sorting. B+ tree.

B. List of Practicals

(This is a suggestive list only. Questions need not be restricted to this list. The practical are advised to be performed in Linux environment using C programming language.)

(a) Write a program to declare an array and initialize the values according to the user. Now ask the user for a number *n* and return the *n*th element from the array.

(b) Write a program to implement array initialized with the numbers divisible by three up to 30. Write a function which accepts the array and return the positions of the even numbers in the array.

(c) Implement linked list in a program by writing functions for the following:
   a. Create a singly linked list of *n* nodes
   b. Count the number of nodes in the list
   c. Print the values of all the nodes
   d. Add a node at first, last and *k*th position in the linked list

e. Delete a node from first, last and k^{th} position

f. Search for an element in the list. If found, return the position of the node. If not found, return a negative value.

(d) Write a program to implement doubly linked list.

(e) Write a function to concatenate two linked lists.

(f) Write a program to take a number *k* and split the linked list after k^{th} position.

(g) Write a program to merge two sorted linked lists.

(h) Write a program to implement list of lists.

(i) Write a program to implement stack using array. Use push and pop operations on the array representation of the stack. Check whether the stack is full or empty.

(j) Write a program to implement stack using linked list. Use push and pop operations on the stack by inserting nodes and deleting nodes from the linked list. Also check if the stack is full or empty.

(k) Write a program to evaluate a simple postfix expression using stack.

(l) Write a program to convert a decimal number into binary number using stack.

(m) Write a program to implement queue using array. Add new elements to the queue and remove elements from the queue represented by array. Check whether the queue is full or empty.

(n) Write a program to implement queue using linked list. Add new elements to the queue and remove elements from the queue represented by linked list. Also check whether the queue is full or empty.

(o) Implement binary search and linear search algorithms on arrays.

(p) Implement binary search tree using array by writing a program to:

a. Create a binary search tree using array
b. Print the prefix notation of the BST
c. Print the infix notation of the BST
d. Print the postfix notation of the BST
e. Search for an element in the BST

(q) Implement binary search tree using linked list by writing a program to:

a. Create a binary search tree using linked list
b. Print the prefix notation of the BST
c. Print the infix notation of the BST
d. Print the postfix notation of the BST
e. Search for an element in the BST
f. Display inorder , preorder and postorder list

(r) Implement following sorting algorithms:

a. Heap sort
b. Quick sort
c. Radix sort
d. Counting sort

(s) Create AVL tree, also delete node from AVL tree

(t) Create Red-BlackAVL tree, also delete node from Red-Black tree

COM0800204: EMBEDDED SYSTEM

1. Learning Outcome:

After completion of this course, students will be able to

1. Understand the basic concepts of embedded system
2. Understand how an entire embedded system is designed
3. Understand how an embedded system is assessed

2. Course outcomes:

After completion of this course, students will be able to

CO1: Identify the basic components of an embedded system

CO2: Analyze the hardware and software aspects of embedded system

CO3: Adopt a systematic approach of development of embedded system

CO4: Understand the basics of validating an embedded system

3. Prerequisite: NIL

4. Semester: 7

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 4

8. Practical Credit: 0

9. Number of required hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non Contact: 5 hrs

10. List of reference books:

a) Peter Marwedel, Embedded System Design, Springer
b) Elecia White, Making Embedded Systems, O′Reilly

11. Detailed Syllabus:

UNIT 1: Introduction (3 lectures)

Definition of Embedded system, Need of embedded systems, challenges in embedded system design, common characteristics of embedded systems

UNIT 2: Specifications and Modeling (15 lectures)

Requirements, Models of computation, StateCharts: Modeling of hierarchy, Timers, Edge labels and StateCharts semantics, Evaluation and extensions, General language characteristics: Synchronous and asynchronous languages, Process concepts, Synchronization and communication, Specifying timing, Using non-standard I/O devices, SDL, Petri nets: Introduction, Condition/event nets, Place/transition nets, Predicate/transition nets, Evaluation, Message Sequence Charts, UML, Process networks, Task graphs, Asynchronous message passing, Synchronous message passing, Java, VHDL, SystemC, Verilog and SystemVerilog, SpecC, Levels of hardware modeling

UNIT 3: Embedded System Hardware (12 lectures)

Components of embedded system hardware, Input: Sensors, Sample-and-hold circuits, A/D-converters, Communication: Requirements, Electrical robustness, Guaranteeing real-time behavior, Examples, Processing Units: Overview, Application-Specific Circuits (ASICs), Processors, Reconfigurable Logic, Memories, Output, D/A-converters, Actuators

UNIT 4: Scheduling and Operating System (10 lectures)

Prediction of execution times, Scheduling in real-time systems: Classification of scheduling algorithms, Aperiodic scheduling, Periodic scheduling, Resource access protocols, Embedded operating systems: General requirements, Real-time operating systems, Middleware: Real-time data bases, Access to remote objects

UNIT 5: Hardware/ Software Codesign (15 lectures)

Task level concurrency management, High-level optimizations: Floating-point to fixed-point conversion, Simple loop transformations, Loop tiling/blocking, Loop splitting, Array folding, Hardware/software partitioning: Introduction, COOL, Compilers for embedded systems: Introduction, Energy-aware compilation, Compilation for digital signal processors, Compilation for multimedia processors, Compilation for VLIW processors, Compilation for network processors,

Compiler generation, retargetable compilers and design space exploration, Voltage Scaling and Power Management: Dynamic Voltage Scaling, Dynamic power management (DPM), Actual design flows and tools, SpecC methodology, IMEC tool flow

UNIT 6: Validation (5 lectures)

Introduction, Simulation, Rapid Prototyping and Emulation, Test: Scope, Design for testability, Self-test programs, Fault simulation, Fault injection, Risk- and dependability analysis, Formal Verification

COM0800304: MOBILE APPLICATION DEVELOPMENT

1. Learning Outcome:

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

- Understand the basics of mobile application development
- Adopt the developmental environment to develop robust applications.
- Understand, write and debug applications.

2. COURSE OUTCOMES:

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

CO1: Outline basic application layout.

CO2: Associate with different application development languages, frameworks and environments.

CO3: Explain functions, intents, and APIs for different applications.

CO4: Apply and design with local and real-time databases.

CO5: Develop a complete applications.

3. Prerequisite: NIL

4. Semester: 8

5. Course Type: Compulsory

6. Course Level: 500

7. Theory Credit: 3

8. Practical Credit: 1

9. Number of required hours:

a) Theory: 45 hrs

b) Practical: 30 hrs

c) Non Contact: 5 hrs

10. List of reference books:

a) B. Phillips, C. Stewart, K. Marsicano, B. Gardner. “Android Programming: The Big Nerd Ranch Guide”, 5thEd., O’Reilly Media, 2022.

b) R. Meier, I. Lake. “Professional Android”, 4th Ed., Wiley, 2018

c) Android Documentation - https://developer.android.com/

11. Detailed Syllabus:

Unit 1: Fundamentals of Mobile Applications (5 Lectures)

Introduction to mobile app development. Overview of different mobile platforms – Android, iOS. Platforms of mobile application development – Native vs Cross-platform. Understanding mobile app design and user experience principles. UI/UX design tools, concept of wire-framing and user research.

Unit 2: Stepping into Mobile App Development (18 Lectures)
Basics of Java or Kotlin. Getting started with Android Studio IDE. Setting up development environments – editor, emulators and debugger. Exploring SDKs. Deploying and Debugging the basic "Hello World" app. Running on emulator, deployment as SaaS or PaaS.

Understanding Android *Activity* and its lifecycle, various events associated with Activity Lifecycle – *onCreate()*, *onStart()*, *onPause()*, *onResume()*, *onStop()*, *onRestart()*, *onDestroy()* etc. Understanding various essential folders and files associated with an Android App stored inside*manifests, java* and *res* directories. Basic understanding about *Gradle*.

Understanding the Mobile Application Development Lifecycle. Exploring app components – Activity, View, Fragments. Using *RecyclerView*, *ListView*, *ImageView* and *WebView*. Understanding and working with fragments. Using of intents – broadcasting and receiving intents. Passing data and objects through intents. Adapting display orientation. Managing notifications, action bar. Basics of action handling. Basics of Model View Controller (MVC) architecture. Building a simple app using MVC pattern.

Unit 3: Working with Database and API (10 Lectures)
Introduction to database for mobile application. Using SQLite for local database. Interacting with Remote Databases using JSON. Using input methods with Input Method Editor (IME). Introduction to Application Programming Interface (API). Using common APIs like Google Maps, YouTube, Open Weather in an application. Features of RESTful API. Integrating RESTful API for designing a CRUD application.
Setting up real-time databases like Firebase. Working with an application with Firebase as backend.

Unit 4: Advanced topics and Deployment (12 Lectures)
Introduction to mobile development frameworks – Flutter, React Native. Introduction to cross-platform mobile application development. Getting started with flutter. Creating and deploying a basic Flutter app in both Android and iOS using Android Studio and Xcode at Andoid and iOS respectively.
Overview of social SDK. Setting permissions – install time, runtime permissions. Exploring and updating application manifest. Publishing into App Stores.

Practical Assignments:

1. Develop a simple Android app using Java/Kotlin that displays "Hello World" on the screen.
2. Set up Android Studio IDE and create an emulator for testing your apps.
3. Create a basic Android app that utilizes different Activity lifecycle methods and displays log messages for each lifecycle event.
4. Implement a RecyclerView in an Android app to display a list of items retrieved from a hardcoded array.

5. Create a multi-pane UI using Fragments for both smartphones and tablets, with navigation between fragments.
6. Develop an app that uses intents to navigate between activities and passes data (e.g., name, age) from one activity to another.
7. Build a CRUD (Create, Read, Update, Delete) app in Android that uses SQLite for local database storage.
8. Integrate Firebase into an Android app to store and retrieve data in real-time, such as a simple chat application.
9. Use JSON parsing to fetch data from a public API (e.g., weather data) and display it in your Android app.
10. Implement Google Maps API in an Android app to display a map with a marker at a specific location.
11. Create a basic Flutter app that displays text and images on both Android and iOS platforms.
12. Explore and implement runtime permissions in an Android app for accessing device features like camera or location.
13. Update the Android app manifest file to include necessary permissions, activities, and services.
14. Add notifications to an Android app using the NotificationCompat API for displaying messages to the user.
15. Implement basic user authentication using Firebase Authentication in an Android app.
16. Develop a simple app using React Native that displays a list of items with basic CRUD functionality.
17. Create a responsive UI for different screen sizes and orientations in both Android and iOS using Flutter.
18. Integrate social media SDK (e.g., Facebook SDK) into your Android app for sharing content.
19. Implement error handling and logging mechanisms in your Android app to capture and report app crashes or exceptions.
20. Prepare your Android app for deployment by generating a signed APK, conducting testing on different devices, and optimizing app performance.

COM0800404: SYSTEM ADMINISTRATION AND NETWORKING

1. **Learning Outcomes**: After successful completion of this course, students will be able to:
    - Understand the roles and responsibilities of system administrators.
    - Explain and manage operating systems, services, and user accounts.
    - Utilize network tools to analyze, diagnose network traffic.

2. **COURSE OUTCOMES**:
    At the end of the course, students will be able to:
    CO1: Describe responsibilities of a system administrator and understand various administration tools.
    CO2: Explain history, uses and significance of GNU/linux, free and open source software.
    CO3: Demonstrate linux administration and networking skills.
    CO4: Propose solutions to system and network failures.
    CO5: Design computer networks of various configurations.
3. **Prerequisites**: **Computer networks**
4. **Semester**: **8**
5. **Course Type**: **Elective**
6. **Course Level**: **500-599**
7. **Theory Credit**: 3
8. **Practical Credit**: 1
9. **No of required hours**:
    a) Theory: 45 hrs
    b) Practical: 30 hrs
    c) Non Contact: 5 hrs
10. **List of Reference Books**:
    a) UNIX and Linux System Administration Handbook, Trent R. Hein, Evi Nemeth, Garth Snyder, Ben Whaley, Dan Mackin, 5th edition, Addison-Wesley
    b) Computer Networking: A Top-Down Approach by James F. Kurose and Keith W. Ross, 6th edition,Pearson Education
    c) Network Warrior, Gary A. Donahue, 2nd edition, O′Reilly

11. **Contents of Syllabus**:

A. Theory

Unit I: Introduction to System Administration (5hrs)

Role and responsibilities of a system administrator, Overview of system administration tools and technologies: vim, nano, wireshark, Clonezilla, PuTTy, FileZilla etc, Introduction to various operating systems (Windows, GNU/Linux, macOS), Introduction to virtualization software: VirtualBox, VMware, Windows Subsystem Linux(WSL) etc.

Unit III : Introduction to GNU/Linux (2hrs)

History and brief overview of GNU/Linux: Free Software Movement, Free software vs Open source software, General Public License (GPL), Linux distribution (distro)

Unit IV : Linux kernel and file system (10hrs)

Major components of the Linux operating systems, Linux kernel : linux kernel architectures, linux kernel features, Linux booting up process: POST(Power-On Self-Test), BIOS(Basic input/output system), MBR(Master boot record)/UEFI(Unified Extensible Firmware Interface), Linux File systems, linux directory structure, Types of files in linux file system.

Unit V: Operating System Administration using linux (10hrs)

Managing users and groups: Create, modify, delete user and group accounts, File ownership of user and group, File access permissions of group and users, Process management : starting, monitoring and terminating processes, File system management: mounting and unmounting file systems, Disk management: monitoring disk usage, Backup and restore in linux.

Unit VI: Networking Fundamentals (10hrs)

Basics of IPv4 and IPv6 addressing, IP address classes : Classful address and Classless Inter-Domain Routing (CIDR) notation, subnet, subnet mask, Network interfaces: ethernet, loopback, wireless etc. IANA(Internet Assigned Numbers Authority) assigned well known ports. Brief overview of The Network Information System, Structure and function of the Domain Name
Service (DNS), DHCP (Dynamic Host Configuration Protocol),

Unit VII: Network Management using linux (5hrs)

Network interface management using ifconfig, ip tools: set, update IP address and subnet mask, bringing interfaces up and down. Usage of diagnostics tools: ping(Packet Internet or Inter-Network Groper), traceroute, Netstat and tcpdump command.

Unit VIII: Network Security (3hrs)

Overview of network security threats and vulnerabilities, Firewalls and intrusion detection/prevention systems (IDS/IPS), VPN (Virtual Private Network) for secure remote access, Encryption techniques (SSL/TLS, IPsec), Secure Shell (SSH)

B. List of practicals:

1. Design a wired LAN network using Cisco Packet Tracer - Networking Simulation Tool.
2. Design a wireless LAN network using Cisco Packet Tracer - Networking Simulation Tool.
3. Design two LAN networks with different IP ranges and connect both networks using a router using Cisco Packet Tracer - Networking Simulation Tool.
4. Design a DHCP server using Cisco Packet Tracer - Networking Simulation Tool.
5. Design an email server using Cisco Packet Tracer - Networking Simulation Tool.
6. Usage of commands ls, cat, pwd, cd, mkdir, man etc.
7. Understand the /etc/passwd file, /etc/shadow, /etc/group file.

8. Manage users and groups (create, delete, modify) using useradd, usermod, userdel, groupadd, groupmod, groupdel commands.
9. Change file ownership using the chown command.
10. Change file access permissions using the chmod command.
11. Usage of process management tools like ps, top, kill etc.
12. Usage of disk utility tools like du, df, mount etc.
13. Usage of backup and restore tools like tar, cpio.
14. Usage of net-tools like ifconfig, ip for assigning, deleting, modifying IP, subnet mask for network interfaces .
15. Analysis of network traffic with diagnostics tools like ping, net-stat, traceroute, tcpdump.

COM0800504: MOBILE COMPUTING

1. **Learning Outcomes:**
   - Learn the basic concepts and principles in mobile computing including major techniques involved, and networks & systems issues for the design and implementation of mobile computing systems and applications.
   - Understand both theoretical and practical issues of mobile computing.
   - Understand the key components and technologies involved and to gain hands-on experiences in building mobile applications.

2. **Course Outcomes**:

On successful completion of course, learner will be able to

CO1: Identify basic concepts and principles in computing, cellular architecture.
CO2: Describe the components and functioning of mobile networking.
CO3: Classify variety of security techniques in mobile network.
CO4: Apply the concepts of WLAN for local as well as remote applications.
CO5: Describe Long Term Evolution (LTE) architecture and its interfaces.

3. Prerequisites: Computer Networks
4. Semester: 8
5. Course Type: Elective
6. Course Level: 500
7. Theory Credit: 4
8. Practical Credit: 0
9. No of Hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non-Contact: 5 hrs

10. List of Books:

a) Jochen Schilller, “Mobile Communication”, Addision wisely, Pearson Education
b) William Stallings “Wireless Communications & Networks”, Second Edition, Pearson Education
c) Christopher Cox, “An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G
d) Mobile Communications”, Wiley publications
e) Raj Kamal, “Mobile Computing”, 2/e, Oxford University Press-New

11. Contents of Syllabus:

Unit 1: Introduction to Mobile Computing 4

Introduction to Mobile Computing, Applications of Mobile Computing, Telecommunication Generations,Cellular systems,Electromagnetic Spectrum, Antenna, Signal Propagation, SignalCharacteristics, Multiplexing, Spread Spectrum: DSSS & FHSS, Co-channel interference, MAC Protocols.

Unit 2: GSM Mobile services 8

GSM Mobile services, System Architecture, Radio interface, Protocols,Localization and Calling, Handover, security (A3, A5 & A8), GPRS system and protocol architecture, GPRS, UTRAN, UMTS core network; Improvements on Core Network, Security.

Unit 3: Mobile Networking 8

Medium Access Protocol, Internet Protocol and Transport layer, Mobile IP: IP Packet Delivery, Agent Advertisement and Discovery, Registration, Tunneling and Encapsulation, Reverse Tunneling, Mobile TCP: Traditional TCP, Classical TCP Improvements like Indirect, TCP, Snooping TCP & Mobile TCP, Fast Retransmit/ Fast Recovery,Transmission/Timeout Freezing, Selective Retransmission

Unit 4: Wireless Local Area Networks 8

Wireless Local Area Networks: Introduction, Infrastructure and ad-hoc network, IEEE 802.11: System architecture, Protocol architecture, Physical layer, Medium access control layer, MAC management, 802.11a, 802.11bstandard, Wi-Fi security: WEP,WPA, Wireless LAN Threats, Securing Wireless Networks, Bluetooth: Introduction, User Scenario, Architecture, protocol stack

Unit 5: Mobility Management 8

Mobility Management: Introduction, IP Mobility, Optimization, IPv6, Macro Mobility: MIPv6, FMIPv6, Micro Mobility: CellularIP, HAWAII, HMIPv6,

Unit 6: Long-Term Evolution (LTE) of 3GPP 9

Long-Term Evolution (LTE) of 3GPP: LTE System Overview, Evolution from UMTS to LTE, LTE/SAE Requirements, SAE Architecture, EPS: Evolved Packet System, E-UTRAN, Voice over LTE (VoLTE),Introduction to LTE-Advanced, Self-Organizing Network (SON-LTE), SON for Heterogeneous Networks, (HetNet), Comparison between Different Generations (2G, 3G, 4G and 5G),Introduction to 5G.

COM0800604: PROJECT AND PRESENTATION

1. **Learning Outcome:**

- Students will recall and describe the problem statement, objectives, and methodologies employed in their project.
- Students will demonstrate an understanding of the technologies explored during the project, explaining their relevance, functionality, and potential applications.
- Students will apply the acquired knowledge and skills to develop a solution or prototype addressing the identified problem, utilizing the chosen technologies effectively.
- Students will critically analyze the project outcomes, assessing the strengths and weaknesses of their approach, and identifying areas for improvement or further exploration.
- Students will synthesize their findings into a coherent dissertation, presenting their research methodology, results, and conclusions while evaluating the implications and significance of their work within the broader context of the field.

2. **COURSEOUTCOMES:**

   Attendees of the course, students will be able to:

   CO1:Students will be able to recall and recognize fundamental concepts, principles, and terminology relevant to the course.

   CO2: Students will demonstrate a comprehensive understanding of theoretical frameworks, methodologies, and models discussed in the course.

   CO3: Students will be able to apply acquired knowledge and skills to solve practical problems, analyze case studies, and develop solutions in real-world scenarios relevant to the course content.

   CO4: Students will demonstrate the ability to synthesize diverse sources of information, integrate multiple perspectives, and propose innovative solutions or designs.

   CO5: Write their findings and analysis in the form of a dissertation.

3. **Prerequisite:** Basic Subject knowledge
4. **Semester: 8**
5. **Course Type: Compulsory**
6. **Course Level: 500**
7. **Theory Credit: 0**
8. **Practical Credit: 4**
9. **Number of required hours:**

   a) Theory: 0 hrs

   b) Practical: 60 hrs

   c) Non Contact: 0hrs

10. COURSE CONTENT:

At the onset of their sixth semester, each student will receive an assignment for a project. Students, either individually or in pairs, will delve into a unique problem under the mentorship of a faculty member from the department. The chosen problem should allow students to delve deeply into one or two specific technologies, fostering a strong understanding and proficiency in those areas upon project completion.

To promote innovation and avoid redundancy, previously tackled problems should be avoided unless they hold exceptional research significance and expansive scope. While application-based problems spurred by specific demands may be considered, simplistic information management systems comprising only a few database tables or data entry forms should be discouraged.

Interdisciplinary collaboration where applicable, enabling students to draw insights from diverse fields and perspectives to enrich their projects will be encouraged. Students also have the option to conduct their projects in collaboration with other institutes or organizations, subject to approval from the relevant institute organization. However, at least one project supervisor must be affiliated with the institute or organization.

Regular progress updates must be reported by meetings with the project supervisor throughout the project duration.

Students should look for opportunities to publish their project findings in academic journals, conferences, or other relevant platforms to disseminate their research outcomes and contribute to the academic community.

Projects must culminate in the submission of a dissertation. Evaluation and presentation of projects will adhere to the regulations outlined in the PG course semester system of G.U., with choice-based credit and grading system.

11. COURSE ASSESSMENT DETAILS:

Internal assessment: seminars, presentations, viva, project implementation

COM0900104: ARTIFICIAL NEURAL NETWORK

1. **Learning Outcomes**: At the end of the course, students will be able to:
   (a) Understand linear and non-linear classification.
   (b) Understand supervised and unsupervised learning.
   (c) Apply ANN algorithm for classification.

2. **Prerequisites**: Fundamentals of Python programming

3. **Semester**: 9

4. **Course type**: Elective

5. **Course level**: 500

6. **Theory credit:** 3

7. **Practical credit**: 1

8. **Number of required hours:**

   a) Theory:35 hrs (35 classes)

   b) Practical: 30 hrs (15 classes)

   c) Non-contact: 10 (hours)

9. **Reference books:**
   1. Fundamental of neural networks, LaureneFausett, Pearson
   2. Elements of Artificial Neural Network, 2ed, Kishan Mehrotra, penram International Publishing (India) Pvt. Ltd.
   3. Principles of Soft Computing, SN Sivanandam, SN Deepa, Wiley India
   4. Neural networks and deep learning,Charu C. Aggarwal, Springer
   5. NPTEL

10. **Course Outcomes**

CO-01 Analyze linear and non-linear classification etc.

CO-02 Understand role of activation function, training, testing.

CO-03 Understand different components of artificial neural network.

CO-04 Understand unsupervised learning.

CO-05 Apply supervised and unsupervised learning algorithm.

11. **Detailed Syllabus:**

A. Theory

Unit I : Introduction to ANN 7 Lectures

Introduction to learning, intelligence, and machine learning, applications of ANN, history of neural network, difference between machine learning and deep learning, challenges of machine learning techniques, Biological neuron, artificial neuron, Typical architecture(Single layer net, layered network, fully connected networks, acyclic network, feed forward networks) difference between discriminative and generative learning, common activation function(identity function, binary step function, ramp function,binary sigmoid, bipolar sigmoid), bias .

Unit II: Pattern classification 8 Lectures

Introduction to classification, clustering, pattern association, minimum distance classifier, Euclidean distance, mahalanobis distance, nearest neighbor rule, K-nearest neighbor rule, linear classifier – two class problem, decision boundary between two different classes, design of separating plane, gradient descent algorithm, multi class classification using SVM. Optimizing loss function: Data loss, regularization loss, quadratic loss function, disadvantages of quadratic loss function, cross entropy loss, linear & logistic regression, softmax classifier.

Unit III: Neural Network 7 Lectures

Neural network, input layer, hidden layer, output layer, weight vector, design of neural network using AND function, OR function, XOR function, single layer network without nonlinearity,single layer network with nonlinearity, multilayer perceptron, perceptron training algorithm, choice of learning rate, feed forward neural network, error function: sum of squared, cross entropy, weight updating rule.

Unit IV: Backpropagation Neural Network 8 Lectures

Introduction, architecture, back propagation learning (single layer multiple output, multilayer multiple output), back propagation learning: hidden layer. Some selected applications. Evaluation of networks(quality of results, Generalizability regularization), setting the parametervalues(initialization of weights, frequency of weight updates, choice of learning rate, momentum, number of hidden layers and nodes, number of samples).

Unit V: Unsupervised learning 5 Lectures

Introduction to unsupervised learning, hamming networks, maxnet, simple competitive learning, k-means clustering algorithm, counterpropagation network, Topologically Organized Networks(SOM).

COM0900204: SOFT COMPUTING

1. **Learning Outcomes**
   - Learn about advanced forms of set theories including fuzzy and rough.
   - Understand genetic algorithms and their applications.
   - Learn different classification and clustering algorithms based on Rough and Fuzzy sets.

2. **Course Outcomes**

   CO1: Recognize the difference between Crisp, Fuzzy and Rough sets.
   CO2: Infer classification and clustering methods based on rough and fuzzy sets.
   CO3: Relate Fuzzy to Rough set theory.
   CO4: Prioritize among fuzzy set theory needs and limitations.
   CO5: Distill different methods of genetic algorithms.

3. **Prerequisites:** Discrete Mathematics or equivalent course

4. **Semester**: 9
5. **Course Type:** Elective
6. **Course Level:** 500
7. **Theory Credit**: 4
8. **Practical Credit:** 0
9. **No of Hours:** 60
10. **List of books:**
    (a) 1. S.N. Sivanandam and S.N. Deepa, Principles of Soft Computing, Wiley India Edition.
    (b) 2. J. Valente de Oliveira, W. Pedrycz, Advances in Fuzzy Clustering and its Applications, John Wiley & Sons, 2007.

11. **Content of syllabus:**

UNIT 1:Fuzzy Sets: Classical(crisp) sets and operations on sets. Fuzzy Sets. Notion of uncertainty of membership in a fuzzy set. Fuzzy set operations and their properties. Classical Relations and Fuzzy Relations. Features of the membership functions, fuzzification, methods of membership value assignments – Intuition, Inference, Rank Ordering and Angular Fuzzy sets. Defuzzification: Introduction, Alpha-Cuts. Defuzzification Methods – Max-membership Principle, Centroid method, Mean-max Membership, Centre of Sums, Centre of Largest Area.

UNIT 2: Rough Sets: Information Systems, Indiscernibility relation, lower and upper approximations; negative and boundary regions of rough sets. Independence of attributes. Core and Reducts of attributes and attribute values. Decision Systems - Dispensable and Indispensable Attributes; Reducts and Cores.

UNIT 3: Fuzzy Clustering: Limitations of hard partitioning and need for fuzzy clustering, Fuzzy c-means (FCM) algorithm.

UNIT 4: Rough Set Based Methods: Information granulation using rough sets, decision rules in rough set models, classification, and clustering methods based on rough sets.

UNIT 5: Genetic Algorithm: What areGenetic Algorithms? Basic terms and terminologies in GA's. Search space. Encoding and Reproduction cycle. Genetic operators – mutation, crossover and selection. Fitness function. Selection methods - Roulette wheel, Stochastic universal sampling, Binary tournament. Elitism. The general Genetic Algorithm with Flowchart. k-means clustering using GA.

COM0900304: BLOCKCHAIN TECHNOLOGY

1. Learning Outcome:

After completion of this course, students will be able to

- Understand the basic concepts of blockchain
- Understand how a blockchain based application works
- Adopt the concept of blockchain in distributed application development

2. Course outcomes:

After completion of this course, students will be able to

CO1: Identify the basic components of a blockchain application
CO2: Analyze the working of blockchain based applications
CO3: Explain the power and limitations of blockchain based application
CO4: Understand the basics of designing DAPP in ethereum platform

3. Prerequisite: NIL

4. Semester: 9

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 4

8. Practical Credit: 0

9. Number of required hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non Contact: 5 hrs

10. List of reference books:

a) Daniel Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress
b) Mastering Ethereum: Building smart contracts and DAPPS

11. Detailed Syllabus:

UNIT 1: Introduction (15 lectures)

Two software architectures: centralized and distributed, advantages of distributed systems, disadvantages of distributed system, distributed peer to peer systems, The Potential of Peer-to-Peer Systems, The Definition of a Peer-to-Peer System, Architecture of Peer-to-Peer Systems, The Link Between Peer-to-Peer Systems and the blockchain, The Potential of the Blockchain, Trust and Integrity in Peer-to-Peer Systems, Integrity Threats in Peer-to-Peer Systems, The Core Problem to Be Solved by the Blockchain, blockchain as data structure, blockchain as algorithm, blockchain as a suite of technologies, blockchain as an umbrella term for purely distributed peer-to-peer systems with a common application area, provisional definition of blockchain, Ownership and Witnesses, Foundations of Ownership, Identification, Authentication and Authorization, purposes and properties of a ledger, ownership and the blockchain, the double spending problem

UNIT 2: How the Blockchain Works (20 lectures)

Documenting ownership with the blockchain: Describing the transfer of ownership, Maintaining the history of transfers, Importance of Ordering, Integrity of the Transaction History, hash functions, properties of cryptographic hash function, SHA-256, patterns of hashing data, Comparing Data, Detecting Changes in Data, Referring to Data in a Change-Sensitive Manner, Storing Data in a Change-Sensitive Manner: chain and tree, markle tree, Causing Time-Consuming Computations, hash puzzles, The Difficulty Level, Cryptography, Symmetric Cryptography, Asymmetric Cryptography, Creating and Distributing the Keys in asymmetric cryptography, Using the Keys, Asymmetric Cryptography in the Blockchain, Digital Signature, Creating a Signature, Verifying Data by Using the Signature, Identifying Fraud by Using the Signature, Digital Signature in blockchain, RSA algorithm, Storing Transactions in the Blockchain Data Structure, Adding New Transactions, Detecting Changes, The Costs of Manipulating the Blockchain Data Structure, Distributing the Data Store Among Peers, purposes of communication among nodes: Keeping existing connections alive, Establishing new connections, Distributing new information, building blocks for Verifying and Adding Transactions: Validation rules, Reward, Punishment, Competition, Peer control, how the process of verification and addition of new transaction works, Consensus algorithms, PoW, PoS, PoB, PBFT, dealing with dishonest behavior, The Role of Fees Within the Blockchain, definition and characteristics of blockchain

UNIT 3: Limitations, types and Applications of Blockchain (10 lectures)

Technical limitations of blockchain, Non Technical limitations of blockchain, Overcoming the limitations, Conflicting goals of blockchain, roots of the conflicts, solving the conflicts, Four versions of blockchain, Consequences of restricting access, revisiting the characteristics of blockchain, generic application patterns, analyzing blockchain applications

UNIT 4: Ethereum and DAPP (15 lectures)

What is ethereum, Bitcoin, Ethereumvsbitcoin, Ethereum: A General-Purpose Blockchain, components of ethereum, The Structure of a Transaction, The Transaction Nonce, Transaction Gas Transaction Recipient, Transaction Value and Data, Special Transaction: Contract Creation, Digital Signatures, The Signature Prefix Value (v) and Public Key Recovery, Separating Signing and Transmission (Offline Signing), Transaction Propagation, Recording on the Blockchain, Multiple-Signature (Multisig) Transactions, What Is a Smart Contract?, Life Cycle of a Smart Contract, Introduction to Ethereum High-Level Languages, Building a Smart Contract with Solidity, The Ethereum Contract ABI, Programming with Solidity: Data Types, Predefined Global Variables and Functions, Contract Definition, Functions, Contract Constructor and selfdestruct

COM0900404: DIGITAL IMAGE PROCESSING

1. **Learning Outcomes:** At the end of the course, students will be able to learn:
   - concept of digital image
   - Image sampling and quantization
   - Image processing in spatial and frequency domain
   - Image segmentation, binary and colour image processing
   - Image compression

2. Prerequisites: Fundamentalsof Python Programming

3. Semester: 9

4. Course type: Elective

5. Course level: 500

6**. Theory credit:** 3

7**. Practical credit**: 1

8. Number of required hours:

a) Theory:35 hrs (35 classes)

b) Practical: 30 hrs (15 classes)

c) Non Contact: 10

9. List of Reference books:

a) R. C. Gonzalez & R. E. Woods - Digital Image Processing, Addison Wesley, 1993.
b) K. Jain - Fundamentals of Digital Image Processing, PHI
c) K. R. Castleman - Digital Image Processing, PHI 1996
d) W. K. Pratt - Digital Image Processing, John Wiley Interscience, 1991
e) Sanjay Sharma – Digital Image Processing, SK Kataria& Sons
f) NPTEL, IITKGP

10. Course Outcomes:

CO-01	Define the scope of digital image processing
CO-02	Understand image acquisition techniques
CO-03	Understand digital image and image segmentation
CO-04	Apply binary and colour image processing
CO-05	Understand image compression

11. Detailed Syllabus:

A. Theory

Unit I : Digital Image processing

Introduction, application of DIP, Image digitization, sampling, quantitation, display, signal reconstruction from samples, convolution concept, signal reconstruction from image, quantizer design, relationship between pixels, basic transform

Unit II : Image transformation

Introduction, Fourier transform, 1D and 2D Fourier transform, Fourier transform in continuous and discrete domain, properties of Fourier transform, FFT, KL transform, Discrete cosine transform, Walsh transform, Handmard transform, Other transforms: Other separable image transforms
and their algorithms.

Unit III : Image enhancement

Enhancement in spital domain, point processing techniques, mask processing, contrast stretching operations, histogram equalization, Spatial Filtering, Frequency Domain Filtering

Unit IV : Image degradation

Frequency domain processing, Image restoration techniques, estimation of degradation model and restoration techniques, image registration, Inverse Filtering, Wiener Filtering.

Unit V :Colour Image processing

Colour fundamentals, different colour models, RGB, CMY, HIS etc.

Unit VI : Image segmentation

Introduction, different approaches for image segmentation, Edge detection, Line detection, Image segmentation: Global processing(Hough Transform), region-based segmentation operation, Thresholding techniques, region splitting and merging

Unit VII : Binary image processing

Binarisation, arithmetic and logical operations, Dilation, Erosion, opening and closing operations, Some basic morphological operations: Boundary extraction, Region filling, Hit-or-Miss transformation, Thinning and ThickeningDistance Transform

Unit VIII : Image compression

Introduction, Types of data redundancies, Image compression model, Lossy Compression, Loss-less compression, Run-length, and Huffman Coding, Transform Coding, LZW compression, Image Compression Standards.

B. Practical

a. Reading and display digital image
b. Image enhancement using spital domain techniques
c. Image segmentation using mask processing and binary image processing techniques
d. Apply image compression technique
e. Drawing histogram for binary and colour image

COM0900504: ADVANCE MACHINE LEARNING TECHNIQUES

1. Learning Outcomes: At the end of the course, students will be able to:

- Understand linear and non-linear classification.
- Understand the role of feature extraction.
- Apply machine learning algorithms to real world problems.

2. Prerequisites: FundamentalsofPythonprogramming&Fundamentals of Machine Learning.

3. Semester: 9

4. Course type: Elective

5. Course level: 500

6. Theory credit: 3

7. Practical credit: 1

8. Number of required hours:

a) Theory:35 hrs (35 classes)

b) Practical: 30 hrs (15 classes)

c) Non- Contact: 10

9. Reference books:

a) Fundamental of Neural Networks, LaureneFausett, Pearson
b) Principles of Soft Computing, SN Sivanandam, SN Deepa, Wiley India
c) Neural Networks and Deep Learning, Charu C. Aggarwal, Springer
d) NPTEL

10. Course Outcomes:

CO-01	Analyzelinearandnon-linearclassificationetc.
CO-02	Understand role of features in classification.
CO-03	Understand different components of artificial neural network.
CO-04	Understand unsupervised learning using autoencoder
CO-05	Apply supervised and unsupervised learning algorithm.

11. Detailed Syllabus:

A. Theory

Unit I :Introduction to machine learning

Introduction to learning, intelligence, and machine learning, applications of machine learning, difference between machine learning and deep learning, challenges of machine learning techniques, feature descriptors, difference between discriminative and generative learning.

Unit II : Feature extraction

Introduction to features, feature space representation(2D), Boundary features, region features: intensity, texture(co-occurrence matrix), colour, Statistical moments, histogram features audio signal features: time domain(magnitude, zero crossing rate, energy), frequency domain(cepstral coefficient, Mel frequency cepstral coefficient), Bayesian learning, discriminant function.

Unit III Linear classifier

Introduction, minimum distance classifier, Euclidean distance ,mohalanobis distance, nearest neighbor rule, K-nearest neighbor rule, linear classifier – two class problem, decision boundary between two different classes, design of separating plane, gradient descent algorithm, multi class classification using SVM. Optimizing loss function: Data loss, regularization loss, quadratic loss function, disadvantages of quadratic loss function, cross entropy loss, linear & logistic regression, softmax classifier.

Unit IV : Optimization

Optimization in machine learning, Stochastic gradient descent, batch, mini batch optimization, advantages and disadventages of each techniques. Underfitting and overfitting, nonlinearity, nonlinear mapping, nonlinear function : threshold, logistic regression, Rectified Linear Unit(ReLU), sigmoid.

Unit V : Neural Network

History of neural network, biological neuron, artificial neuron, bias, neural network, input layer, hidden layer, output layer, weight vector, design of neural network using AND function, OR function, XOR function, single layer network without nonlinearity,single layer network with nonlinearity , multi layerpreceptron, feed forward neural network, error function: sum of squared, cross entropy, weight updation rule, back propagation learning(single layer multiple output, multi layer multiple output), back propagation learning: hidden layer.

Unit VI : Autoencoder

Applications of autoencoder, Unsupervised learning, label and unlabel data, disadvantages of unsupervised learning, neural network as unsupervised learning, structure of autoencoder, input layer, bottleneck layer, output layer, under complete autoencoder, stacked autoencoder, deep autoencoder, introducation to principal component analysis(PCA), relation between autoencoder vs. PCA, training of autoencoder, introduction to sparse autoencoder, denoising autoencoder.

COM0900604: BIG DATA ANALYTICS

1. **Learning Outcome:**
- Student will be able to understand several key big data technologies.
- Student will be able to learn about concept of machine learning in Big Data analysis.
- Student will be have concept of Hadoop framework and MapReduce.

2. **COURSEOUTCOMES:**

   Attendees of the course, students will be able to:

   CO1: Identify key concepts, techniques, and tools used in Big Data analytics.

   CO2: Demonstrate an understanding of the underlying principles and algorithms employed in Big Data analytics.

   CO3: Apply their knowledge of Big Data analytics tools and techniques to analyze and interpret real-world datasets.

   CO4: Analyze the results of their data analyses, assessing the accuracy, reliability, and validity of the insights obtained.

   CO5: Identifying potential biases or limitations in the data and analytical methods.

3. **Prerequisite:** NIL
4. **Semester: 9**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 3**
8. **Practical Credit: 1**
9. **Number of required hours:**

   a) Theory: 45 hrs

   b) Practical: 30 hrs

   c) Non Contact: 5 hrs

10. **List of reference books:**

    a. Michael Minelli, Michehe Chambers, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Business", 1st Edition, AmbigaDhiraj, Wiley CIO Series, 2013

    b. Bart Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and Its Applications, Wiley

    c. Tom White, "Hadoop: The Definitive Guide", 3rd Edition, O'reilly, 2012

    d. Han, J., Kamber, M., Pei, J. (2009). Data mining: Concepts andtechniques (3rd ed.). Morgan Kaufmann.

    e. Kai Hwang, Min Chen, Big-Data Analytics for Cloud, IoT and Cognitive Computing, Wiley

    f. ArvindSathi, Big Data Analytics: Disruptive Technologies for Changing the Game, MC Press

11. **Detailed Syllabus:**

UNIT 1: Introduction to Big Data (10 Lectures)

A Flood of Mythic "Start-Up" Proportions, Big Data Is More Than Merely Big, four V's, A Convergence of Key Trends, A Wider Variety of Data, The Expanding Universe of Unstructured Data, Digital Marketing and the Non-line World, Database Marketers, Pioneers of Big Data, Big Data and the New School of Marketing, Fraud and Big Data, Credit Risk Management, Big Data and Algorithmic Trading, Big Data and Advances in Health Care, Advertising and Big Data, Using Consumer Products as a Doorway.

UNIT 2: Big Data Technology (8 Lectures)

Hadoop's Parallel World, Data Discovery, Open-Source Technology for Big Data Analytics, The Cloud and Big Data, Predictive Analytics, Software as a Service BI, Crowd-sourcing Analytics, Inter- and Trans-Firewall Analytics, Information Management,

UNIT 3: Data Collection, Sampling, and Preprocessing (10 Lectures)

Types of Data Sources, Sampling, Types of Data Elements, Visual Data Exploration and Exploratory, Missing Values, Outlier Detection and Treatment, Standardizing Data, Categorization, Weights of Evidence Coding, Variable Selection, Segmentation

UNIT4: Predictive and Descriptive Analytics (10 Lectures)

Target Definition, Linear Regression, Logistic Regression, Decision Trees, Neural Networks, Support Vector Machines, Ensemble Methods, Multiclass Classification Techniques, Evaluating Predictive Models, Association Rules, Sequence Rules

UNIT 5: Hadoop (7 Lectures)

A Brief History of Hadoop, Apache Hadoop and the Hadoop Ecosystem, MapReduce, The Hadoop Distributed Filesystem, The Design of HDFS, HDFS Concepts, The Command-Line Interface, HadoopFilesystems, The Java Interface, Data Flow, Parallel Copying with distcp, Hadoop Archives

COM0900704: BIOINFORMATICS

1. **Learning Outcome:**

   Student will be able tounderstand number of biological concepts

   a) Student will be able to learn about vital aspect of bioinformatics databases.
   b) Student will be able to illustrate the effectiveness of sequence alignments.
   c) Student willhave abroad understanding of computational techniques and resources available to biological scientists.
   d) Student will be able summarize techniques for aligning DNA and protein sequences together.
   e) Student will be able to understand genomic sequences.

2. **COURSE OUTCOMES:**

   Attendees of the course, students will be able to:
   CO1: Explain how DNA stores genetic information
   CO2: Apply algorithms for searching biological databases
   CO3: Describe different uses of protein and DNA sequence alignments.
   CO4: Implement phylogenetic tree construction algorithms.
   CO5: Analyse gene and protein secondary structures.
3. **Prerequisite: NIL**
4. **Semester: 9**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **Number of required hours:**

   a) Theory: 60 hrs
   b) Practical: 0 hrs
   c) Non Contact: 5 hrs

10. **List of reference books:**

    a) MarketaZvelebil& Jeremy O. Baum, *Understanding Bioinformatics*; Garland Science, Taylor & Francis Group
    b) Phillip Compeau, PavelPevzner, *Bioinformatics Algorithms: an Active Learning Approach*
    c) Richard Durbin, Sean R. Eddy, Anders Krogh, Graeme Mitchison; *Biological Sequence Analysis*. Cambridge University Press, 1998.

11. **Detailed Syllabus:**

    UNIT 1: Background Basics (10 Lectures)

    The structure of DNA and RNA, DNA, RNA, and Protein: The Central Dogma, Gene Structure and Control,The Tree of Life and Evolution, Protein structure: Primary and

Secondary Structure, Implication for Bioinformatics, Dealing with databases: The Structure of Databases, Types of Database, Looking for Databases, Data Quality.

UNIT 2: Sequence Alignments (10 Lectures)

Principles of Sequence Alignment, Scoring Alignments,Substitution Matrices, Inserting Gaps, Types of Alignment, Searching Databases, Searching with Nucleic Acid or Protein Sequences, Protein Sequence Motifs or Patterns, Searching Using Motifs and Patterns, Patterns and Protein Function, Substitution Matrices and Scoring, Dynamic Programming Algorithms, Indexing Techniques and Algorithmic Approximations, Alignment Score Significance, Aligning Complete Genome Sequences, Profiles and Sequence Logos, Profile Hidden Markov Models, Aligning Profiles.

UNIT 3: Evolutionary Processes (12 Lectures)

The Structure and Interpretation of Phylogenetic Trees, Molecular Evolution and its Consequences, Phylogenetic Tree Reconstruction, Evolutionary Models and the Calculation of Evolutionary Distance, Generating Single Phylogenetic Trees, Generating Multiple Tree Topologies, Evaluating Tree Topologies,Assessing the Reliability of Tree Features and Comparing Trees,

UNIT4: Genome Characteristics (10 Lectures)

Preliminary Examination of Genome Sequence, Gene Prediction in Prokaryotic Genomes, Gene Prediction in Eukaryotic Genomes, Splice Site Detection, Prediction of Promoter Regions, Confirming Predictions, Genome Annotation, Detection of Functional RNA Molecules Using Decision Trees, Algorithms for Gene Detection in Prokaryotes, Features Used in Eukaryotic Gene Detection, Predicting Eukaryotic Gene Signals, Predicting Exon/Intron Structure

UNIT 5: Secondary Structure (12 Lectures)

Types of Prediction Methods, Training and Test Databases, Assessing the Accuracy of Prediction Programs, Statistical and Knowledge-Based Methods, Neural Network Methods of Secondary Structure Prediction, Prediction of Transmembrane Protein Structure, RNA Secondary Structure Prediction, Secondary Structure and Prediction Accuracy, Secondary Structure Prediction based on: Residue Propensities, The Nearest-Neighbor Methods , Neural Networks,Hidden Markov Models

UNIT 6: Cells and Organisms (6 Lectures)

Analysis of Large-scale Gene Expression, Analysis of Large-scale Protein Expression, Clustering Methods and Statistics -Data Preparation; Cluster Analysis; Clustering Methods

COM1000104: OPTIMIZATION TECHNIQUES

1. **Learning Outcomes**
   - Understand the need for optimization.
   - Learn and apply different optimization techniques.
   - Learn among different algorithms used for optimizations.

   Course Outcomes

   CO1: Recognize the need for optimization and different techniques available.
   CO2: Appraise between different optimization techniques.
   CO3: Illustrate constrained and unconstrained optimization techniques.
   CO4: Solve Linear Programming problems using different methods.
   CO5: Understand the constrained convex optimization problems.

3. **Prerequisites:**
4. **Semester: 10**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **No of Hours: 60**
10. **List of books:**

    1. Gillette, B.G. : Introduction to operations research - A Computer oriented algorithmic approach, McGraw Hill.
    2. N.S. Kambo: Mathematical Programming Techniques; January 2008; EWP.
    3. K. V. Mital: Optimization Methods, Wiley Eastern; 3 rd Edition.
    4. G. Hadley: Linear Programming, Narosa Publications
    5. C. H. Papadimitriou and K. Steiglitz: Combinatorial Optimization - Algorithms and Complexity,
    Prentice Hall.

11. **Content of syllabus:**

UNIT 1: Unconstrained optimization:

Necessary and sufficient conditions for optima, convex sets, convex functions, optima of convex functions, steepest descent, Newton and quasi Newton methods, conjugate direction methods.

UNIT 2: Constrained optimization:

Linear Programming - Mathematical model, Basis, feasible solutions and basic feasible solutions, Graphical solution method, unboundedness, Simplex method, Revised simplex method, Applications, Duality, Dual simplex method, Primal Dual Algorithms. Complexity of the algorithms studied. Ellipsoid Method, Karmakar's algorithm.

UNIT 3: Special models of linear programming problems:

Transportation and assignment problems, Maxflow and shortest path problems, Ford and Fulkerson algorithm, Djikstra's algorithm. Integer programming: Introduction, Travelling Salesman Problem (TSP), Branch and Bound techniques.

UNIT 4: Constrained Convex Optimization:

Problem definition, Kuhn-Tucker Conditions and projected gradient methods.

COM1000204: DATA SCIENCE

1. **Learning Objectives:**
   - Understand the fundamental concepts and methods of data science.
   - Gain proficiency in data manipulation, visualization, and analysis.
   - Learn how to apply statistical techniques to extract insights from data.
   - Develop skills in machine learning and predictive modeling.
   - Learn some advanced techniques used in data science.

2. **Course Outcomes:** Students will be able to-

   CO1: Identify the responsibilities required for a data scientist.

   CO2: Differentiate between types of data and understand the importance of data cleaning and manipulation.

   CO3: Compute statistical summaries and descriptive statistics to understand the underlying patterns in data.

   CO4: Illustrate about text mining, deep learning, and optimization techniques.

   CO5: Implement linear regression, logistic regression, various classification and clustering algorithms, PCA, LDA

3. **Prerequisite: NIL**
4. **Semester:10**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **Number of required hours:**

   a) Theory: 60 hrs
   b) Practical: NIL
   c) Non Contact: 5 hrs

10. **List of reference books:**

   a) "*Python for Data Science Handbook*" by Jake VanderPlas

   b) "*Python for Data Analysis*" by Wes McKinney

   c) "*Introduction to Statistical Learning*" by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

   d) "*Data Science from Scratch*" by Joel Grus

11. **Detailed Syllabus:**

UNIT I: Introduction to Data Science (04 Lectures)

Overview on data science, Role of data scientists, Introduction to data analysis tools and libraries (e.g., Python, Pandas, Matplotlib).

UNIT II: Data Preprocessing (08 Lectures)

Types of data, Data cleaning and manipulation, Handling missing values, Handling outliers, Feature engineering.

UNIT III: Exploratory Data Analysis (08 Lectures)

Data visualization techniques, Statistical summaries and descriptive statistics, Probability distributions and random variables, Correlation analysis, Hypothesis testing.

UNIT IV: Machine Learning Basics (20 Lectures)

Overview of supervised and unsupervised learning, Linear regression and logistic regression, Classification algorithms- K-Nearest Neighbor, Decision tree, Clustering algorithms- K-Means, PAM, DBSCAN, BIRCH.

UNIT V: Model Evaluation and Validation (04 Lectures)

Cross-validation techniques, Evaluation metrics for classification, Evaluation metrics for clustering, Evaluation metrics for regression, Overfitting and Underfitting.

UNIT VI: Feature Selection and Dimensionality Reduction (08 Lectures)

Techniques for feature selection, Dimensionality reduction- Principal Component Analysis (PCA), LDA, t-SNE.

UNIT VII: Advanced Topics in Data Science (08 Lectures)

Introduction to Text mining- TF, IDF, vector space model, Introduction to ANN- different types of ANN models, Introduction to CNN- different types of CNN models like VGG-16, LSTM, Introduction to optimization techniques like genetic algorithm.

COM1000304: DEEP LEARNING

1. **Learning Outcomes:** At the end of the course, students will be able to:
    - (a) Understand difference between machine learning and deep learning.
    - (b) Understand different deep learning algorithms.
    - (c) Apply deep learning algorithms for classification.

2. **Prerequisites:** Introduction to Machine learning and FundamentalsofPythonprogramming
3. **Semester:** 10
4. **Course type:** Compulsory
5. **Course level:** 500
6. **Theory credit:** 3
7. **Practical credit:** 1
8. **Number of required hours:**
    - a) Theory:35 hrs (35 classes)
    - b) Practical: 30 hrs (15 classes)
    - c) Non Contact: 10
9. **Reference books:**
    1. Fundamental of neural networks, LaureneFausett, Pearson
    2. Principles of Soft Computing, SN Sivanandam, SN Deepa, Wiley India
    3. Neural networks and deep learning ,Charu C. Aggarwal, Springer
    4. NPTEL

10. **Course Outcomes**

CO-01	Understand machine learning and deep learning.
CO-02	Define different optimization techniques in deep learning
CO-03	Understand different components of neural network
CO-04	Understand different unsupervised neural network
CO-05	Solve some real time problem using advanced deep learning models

11. Detailed Syllabus:

A. Theory

Unit I : Introduction to deep learning

Introduction to learning, intelligence, and machine learning, applications of deep learning, difference between machine learning and deep learning, challenges of deep learning techniques, difference between discriminative and generative learning, linear time invariant (LTI) system, linear shift invariant(LSI) system, convolution.

Unit II : Optimization

Optimization in machine learning, Stochastic gradient descent, batch, mini batch optimization, advantages and disadventages of each techniques. Underfitting and overfitting, nonlinearity, nonlinear mapping, nonlinear function : threshold, logistic regression, Rectified Linear Unit(ReLU), sigmoid.

Unit III : Neural Network

History of neural network, biological neuron, artificial neuron, bias, neural network, input layer, hidden layer, output layer, weight vector, design of neural network using AND function, OR function, XOR function, single layer network without nonlinearity,single layer network with nonlinearity , multi layerpreceptron, feed forward neural network, error function: sum of squared, cross entropy, weight updation rule, back propagation learning(single layer multiple output, multi layer multiple output), back propagation learning: hidden layer.

Unit VI : Autoencoder

Applications of autoencoder, Unsupervised learning, label and unlabel data, disadvantages of unsupervised learning, neural network as unsupervised learning, structure of autoencoder, input layer, bottleneck layer, output layer, under complete autoencoder, stacked autoencoder, deep autoencoder, introducation to principal component analysis(PCA), relation between autoencoder vs. PCA, training of autoencoder, introduction to sparse autoencoder, denoising autoencoder.

Unit V : Convolutional Neural Network

Convolution operation, 1D and 2D operation in continuous and discrete domain, finite convolution kernel, feature map, padding, stride, typical architecture of CNN, nonlinearity, pooling, fully connected layer, depth of network, 3D convolution, different CNN models: LeNet 5, IMAGENET Large Scale Visual Recognition Challenge(ILSVRC)

COM1000404: DISTRIBUTED SYSTEM

1. **Learning Outcome:**
   - To provide students the concepts of basic architecture and components of distributed systems
   - To familiarize the students with the concepts of various distributed algorithms required for achieving mutual exclusion, leader election and agreements.
   - To give students the concepts of inter-process communication and distributed file system handing

2. **COURSE OUTCOMES:**

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

- CO1: Explain the concept and challenges for designing a distributed system.
- CO2: Describe the architecture and different system models of distributed systems.
- CO3: Analyze different synchronization algorithms used in distributed systems.
- CO4: Illustrate different algorithms used for achieving mutual exclusion and agreements in distributed applications.
- CO5: Explain the concept of Inter-process communication techniques in distributed systems.

3. **Prerequisites: NIL**
4. **Semester: 10**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **No of Hours:**
   a) Theory: 60hrs
   b) Practical: 0hrs
   c) Noncontact: 5hrs
10. **List of Books:**
    a) Tanenbaum & Steen; (2004); Distributed Systems Principles and Paradigms; Pearson Education
    b) Coulouris, Dollimore&Kindberg; (2006); Distributed Systems Concepts and Design; Pearson Education
    c) Udit Agarwal; (2016); Distributed Systems; S. K. Kataria& Sons

11. Contents of Syllabus:

UNIT-1 Introduction to Distributed Systems (6 hours)

Definition of a distributed system, Characteristics of distributed and centralized systems, examples of distributed systems: *internet, mobile and ubiquitous computing*, Difference between computer network and a distributed system, Challenges in distributed systems: *heterogeneity, openness, security, failure handling, concurrency* and *transparency.*

UNIT-2 System Models (8 hours)

Problems faced by designers of a distributed system, types of system models: *Physical model, architectural model* and *fundamental model.* Architectural model: software layers, software architectures (*client-server, peer-to-peer*), variations of client-server architectures, design requirements for distributed architectures. Fundamental models:definitionof fundamental model, concept of interaction model, factors effecting interaction of processes, variations of interaction model, Failure model, types of failures-*omission failures, arbitrary failures* and *timing failures*, failure masking. Security model-protecting objects, securing processes and their interactions, secure channels, uses of security models.

UNIT-3 Synchronization (12 hours)

Inherent limitations of a distributed system, external and internal clock synchronization, physical and logical clocks, Cristian clock algorithm, Berkeley clock algorithm, Network Time Protocol, Causal order of events, happened before relationship between events, Lamport's logical clocks, limitations of Lamport's clock. Vector clocks, Synchronization of events using vector clocks, Causal Order of messages and Birman-Schiper-Stephension protocol. Definition of local states and global states, consistent, inconsistent and strongly consistent global states, ChandyLamport snapshot global state recording algorithm, need of termination detection, Haung's termination detection algorithm.

UNIT-4 Distributed Mutual Exclusion (8 hours)

Requirements of Mutual Exclusion algorithms, Performance measurement metrics, Classification of mutual exclusion algorithm, Token based algorithms, non-token-based algorithm, Central Server Algorithm, Lamport's timestamp algorithm, Ricart-Agrawala algorithm, Raymond's Tree-based algorithm, Election algorithms- the Bully algorithm.

UNIT-4 Agreement Protocols (6 hours)

Definition of agreement in distributed system, System models and aspects of recognizing the agreement protocols, classification of agreement problems (Byzantine, Consensus, Interactive), Relation among agreement protocols, Solutions to the Byzantine agreement problem: *upper bound of number of faulty processes* and *impossibility results*, Applications of agreement algorithms,

UNIT-5 Distributed Scheduling (4 hours)

Distributed scheduler, issues in distributed load distribution, components of load distribution algorithm, need of load distribution algorithms, classification of load distribution algorithms-*sender-initiated, receiver-initiated* and *symmetricallyinitiated*, performance, task migration.

UNIT-6 Interprocess Communication (8 hours)

Introduction to Inter-process Communications, Communication patterns-*Client-serve communication* vs *Group communication,* Characteristics of interprocess communication: synchronous and asynchronous communication, reliability, message destination. Sockets, UDP datagram communication: issues in datagram communication, use of UDP. TCP stream communication: abstraction provided by the API to TCP protocol. External data representation and marshalling, CORBA's common data representation (CDR), remote object reference. Client-server communication, concept of RPC and RMI, Implementation of group communication-IP multicast.

UNIT-7 Distributed File systems (8 hours)

Introduction: characteristics of file systems, distributed file system requirements, Architecture of distributed file system, Desirable features of a good distributed file system, structured vs unstructured files, mutable vs immutable files. Mechanism for building DFS, design issues of a DFS, dependency factors of file accessing models, file caching schemes.

COM1000504:SPEECH PROCESSING

1. Learning Outcomes: At the end of the course, students will be able to:

1. To provide the concepts of basic models for speech production and perception
2. To develop time andf requency domain techniques for estimating speech parameters
3. To introduce students a predictive technique for speech recognition.
4. To provide students the process of speech recognition, synthesis and Prosody modeling.

2. Prerequisites: Knowledge of Python programming

3. Semester: 10

4. Course type: Elective

5. Course level: 500

6. Theory credit: 3

7. Practical credit: 1

8. Number of required hours:

a) Theory:35hrs (35 classes)

b) Practical: 30 hrs (15 classes)

c) Non Contact: 10

9. **Reference books:**

a) L. Rabinerand B. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1995, ISBN0-13- 015157-2

b) L.R. Rabinerand R.W.Schafer, Digital Processing of Speech Signals, Prentice-Hall, 1978, ISBN0-13- 213603-1.

c) J.LFlanagan: Speech Analysis Synthesis and Perception-2nd Edition-Springer Vertag,1972.

d) NPTEL

10. Course Outcomes

CO-01 Explain basic characteristics of speech signal in relation to production and hearing of speech by humans.

CO-02 Analyze different time and frequency domain algorithms of speech feature extraction.

CO-03 Understand speech synthesis

CO-04 Understand trends in developments of automatic speech recognition

CO-05 Explain Test-To-Speech(TTS) system

11. Detailed Syllabus:

A. Theory

UNIT-I: The Speech Signal

The process of speech production and perception in human beings, speech production process, representing speech in time and frequency domain, speech sounds and features: vowels, diphthongs, semivowels, nasal consonants, unvoiced fricative, voice fricative, voiced and unvoiced stops, approaches to automatic speech recognition by machine: acoustic phonetic approach, pattern recognition approach, artificial intelligence approach, source filter model of speech, manners and place of articulation, speech perception, auditory system, human ear

UNIT-II: Speech features

Perception of sound, physical dimension of sound, threshold of hearing, Formant frequency, time domain methods in speech processing, short time energy, short time average magnitude, short time zero crossing rate, short time autocorrelation, Linear predictive coding, segmental and supra segmental features, frequency domain in speech processing: filter bank analysis, short time spectral analysis, cepstral transfer coefficient(CC), mel frequency cepstral coefficients and its variants, fundamental frequency, fundamental frequency characteristics, fundamental frequency extraction methods

UNIT-III: Text To Speech Synthesis

Introduction, steps in text to speech synthesis system, text normalization, text processing, speech synthesis markup language, grapheme to phoneme(G2P), G2P methods, prosody, TTs synthesis models: articulatory, parametric, concatenative, W3C standards, prosody modeling, prosodic structure,

UNIT-IV: Isolated Speech recognition using machine learning

Introduction, machine learning, deep learning, linear classification, multi class classification problem, biological neuron, artificial neuron, activation function, loss function, optimizers, designing ANN, speech recognition using any one techniques(SVM, KNN, ANN, CNN, LSTM)

B**. List of Practicals**

(This is a suggestive list only. Questions need not be restricted to this list. The practical are advised to be performed in Linux environment using C programming language.)

1. Record speech signal for 1 sec duration in 8KHz and 16 KHz(Both mono and stero)

2. Write program to find ZCR, LPC, CC, MFCC for speech signal

3 Design SVM classifier for binary and multiclass classification.

4. Design ANN for binary and multiclass classification.

5 Design CNN for multiclass classification.

COM1000604: GEOGRAPHICAL INFORMATION SYSTEM

1. **Learning Outcomes:**

- Develop an understanding of the world's quickly-growing spatial data infrastructure and of how to put it to work for producing location-based information.
- Identify the relevant spatial characteristics of diverse application areas enabling professionals to integrate spatial thinking and GIS analysis into their careers.
- Have an ability to use geospatial technologies to gain a significant advantage in the information technology field.

2. **Course Outcomes**:

On successful completion of course, learner will be able to

CO1: Identify basic concepts of data input, mapping and output related to GIS.

CO2: Describe the selection, querying and spatial joins of GIS data.

CO3: Explain spatial data analysis and interpretation of this.

CO4: Generalize the nature of spatial databases, including data management.

CO5: Integrate coordinate systems, georeferencing and projections.

3. Prerequisites: Computer Networks

4. Semester: 10

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 4

8. Practical Credit: 0

9. No of Hours:

a) Theory: 60hrs

b) Practical: 0 hrs

c) Non-Contact: 5 hrs

10. List of Books:

a) Geographic Information Systems: An Introduction by Jeffrey Star, J. E. Estes, 2007.

b) An Introduction to Geographical Information Systems (4th Edition) by Ian Heywood, Sarah Cornelius and Steve Carver, 2012.

c) Introduction to Geographic Information Systems by Chang Kang-tsung (Karl), 2016.

d) Learning Geospatial Analysis with Python: Understand GIS Fundamentals and Perform Remote Sensing Data Analysis Using Python 3.7, by Joel Lawhead 3rd Edition, 2019.

11. Contents of Syllabus:

Unit 1: Introduction to Geographic Information Systems 10 hrs

Components of GIS, Physical topology, Spatial and non-spatial data, Geospatial Data types: vector, raster, 3d, networks, Map layers, types, properties and design principles, Data model compression techniques.

Unit 2: Spatial Data Analysis 20 hrs

Pre-processing of spatial datasets, Spatial Interpolation Techniques, Map projections (small and large scale, conformal versus equivalent projections, spatial reference data), Basemapdata sources (physical surface, environmental, political, populations, biology/ecology), Projected coordinate systems (geographic/spherical versus rectangular, spheroids)

Unit 4: Geoprocessing 8 hrs

Geoprocessing overview, Attribute proximity selections, Geocoding, NoData in Raster, symbolize raster maps, hillshade maps raster-based risk index.

Unit 6: Geodatabases 7 hrs

Geodatabase topology, Geodata formats and metadata, geoDB domain and subtypes, tables and attributes, Time in GIS.

Unit 7: Digital Elevation Model and Triangulated Irregular Network 15 hrs

Integration of DEMs with satellite data, Slope and aspect in DEM, DEMs derivatives, Shaded relief models and their applications, DEM based Surface Hydrologic Modelling, DEMs Sources, Applications of DEMs.

COM1000704: NATURAL LANGUAGE PROCESSING

1. **Learning Outcomes:** After completing this course, students will know:
   - Fundamentals of Natural Language Processing.
   - Understanding the representation of Languages.
   - Fundamental understanding of.Natural Language Processing approaches and strategies.
   - Understanding of various algorithms and techniques for text-based processing of Natural Languages.
   - Basic idea about ongoing NLP research and development trends.
2. **Course Outcome:**

   CO1: Understanding the basic concepts of natural language processing

   CO2: Understanding the phonemes of a natural language

   CO3: Analyzing grammatical constitution of sentences in natural language

   CO4: Acquiring the art of processing of natural language speech

3. **Prerequisites:** Data Structures, Python Programming
4. **Semester: 10**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **No of Hours:**

   a) Theory: 45 hrs

   b) Practical: 30hrs

   c) Non-Contact: 5hrs
10. **List of Books:**

    (a) Daniel Jurafsky, James H. Martin. ”Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition”. Prentice-Hall

    (b) Manning, Christopher, and HinrichSchutze. ”Foundations of Statistical Natural Language Processing”. MIT Press.

    (c) NitinIndurkhya, Fred J. Damerau “Handbook of Natural Language Processing”, CRC Press.

    (d) Steven Bird, Ewan Klein, and Edward Loper “Natural Language Processing with Python”, O'Reilly Media.

11. Contents of Syllabus:

Unit I:Fundamentals of NLP 4 hrs

Fundamentals of NLP – Definition and Introduction to Natural Language Processing, Stages/Phases of NLP, Challenges associated with NLP – ambiguity; errors; noisy channel etc. Applications of NLP.Basics of Linguistics; Probability Statistics. Regular Expressions and Finite State Automata

Unit II: 10hrs

Survey of English Morphology - Inflectional Morphology & Derivational Morphology, Finite-State Morphological Parsing, Finite-State Lexicon, Finite State Transducers, FST for Morphological Parsing, Lexicon-Free FSTs: The Porter Stemmer, Word and Sentence Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance. N-Gram Language Model. Smoothing.

Unit III: 12 hrs

Part of Speech Tagging – English Word Classes, Tagsets for English, Part-of-Speech Tagging, Rule-Based Part-of-Speech Tagging, Hidden Markov Model (HMM), HMM Part-of-Speech Tagging, Transformation-Based Tagging, Evaluation and Error Analysis, Advanced Issues in Part-of-Speech Tagging. Hidden Markov and Maximum Entropy Models.

Unit IV: 15hrs

Constituency, Context-Free Rules and Trees, Sentence-Level Constructions, The Noun Phrase, The Verb Phrase and Subcategorization, Auxiliaries, Treebank (The Penn Treebank Project), Spoken Language Syntax, Grammar Equivalence & Normal Form, Finite State & Context-Free Grammars, Grammars & Human Processing,

Parsing as Search, Top-Down and Bottom-Up Parsing with comparison, A Basic Top-down Parser, Adding Bottom-up Filtering, Problems with the Basic Top-down Parser, Left Recursion. Statistical Parsing.

Ambiguity, Syntax-Driven Semantic Analysis, Attachments for a Fragment of English, Idioms and Compositionality, Robust Semantic Analysis, WordNet: A Database of Lexical Relations, Reference Resolution, Text Coherence, Discourse Structure, Psycholinguistic Studies of Reference and Coherence.

Unit V: Applications and Recent Trends 4 hrs

Text Summarization, Optimization based approaches for Summarization, Summarization evaluation, Text classification. Sentiment Analysis and Opinion Mining. Machine Translation, Recent trends in NLP,

12. Practical Assignments:

1. Tokenization: Write a Python script to tokenize a given text into words and sentences using NLTK.
2. Stopword Removal: Remove stopwords from a given text using NLTK's stopwords corpus and word tokenization.
3. Stemming and Lemmatization: Implement stemming and lemmatization on a set of words using NLTK's Porter and WordNetlemmatizers.
4. Part-of-Speech (POS) Tagging: Use NLTK to perform part-of-speech tagging on a given sentence.
5. Named Entity Recognition (NER): Apply NLTK's named entity recognition function to identify named entities in a given text.
6. Frequency Distribution: Calculate the frequency distribution of words in a text document using NLTK's FreqDist() function.
7. WordNet: Explore WordNet using NLTK to find synonyms, antonyms, hypernyms, hyponyms, and entailments for a given word.
8. Basic Text Classification: Implement a basic text classifier using NLTK and a simple feature representation (e.g., bag-of-words) for sentiment analysis or topic classification.
9. Evaluate the performance of NLTK's named entity recognition (NER) system on a standard dataset such as the CoNLL 2003 dataset and compare it with other NER systems.
10. Implement dependency parsing using NLTK's DependencyGraph module to analyze the syntactic structure of sentences and extract grammatical relationships between words.

COM1000804: REMOTE SENSING

1. **Learning Outcomes:** At the end of the course, students will be able to learn:
   (a)History and Applications of Remote Sensing
   (b) Remote sensing Data acquisition
   (c) Differentiate digital data and remote sensing data
   (d) Multispectral and Hyperspectral data.

2. **Prerequisites:** Fundamentals of Image Processing

3. **Semester:** 10

4. **Course type:** Elective

5. **Course level:** 500

6. **Theory credit:** 3

7. **Practical credit**: 1

8. **Number of required hours:**

   a) Theory:35 hrs (35 classes)

   b) Practical: 30 hrs (15 classes)

   c) Non-contact: 10 hrs.

9. **Reference books:**
   a) Introduction to Remote Sensing, 5th ed, James B Campbell., The Guilford Press, New York, London.
   b) Geological Remote Sensing: An Overview, China University of Geosciences.
   c) NPTEL

10. **Course Outcomes**

CO-01	Define the scope of remote sensing
CO-02	Understand image acquisition techniques
CO-03	Understand digital image and remote sensing image
CO-04	Understand satellite system
CO-05	Understand multispectral and hyperspectral images

11. Detailed Syllabus:

A. Theory

Unit I : History and Scope of Remote Sensing
Introduction, definitions, Milestones in the History of Remote Sensing, Overview of the Remote Sensing Process, applications of Remote Sensing, ElectromagneticRadiation, The Electromagnetic Spectrum, Major Divisions of the Electromagnetic Spectrum, RadiationLaws, InteractionswiththeAtmosphere, InteractionswithSurfaces.

Unit II : Image acquisition
Mapping Cameras: Introduction, Fundamentals of the Aerial Photograph, Geometry of the Vertical Aerial Photograph, Digital Aerial Cameras, Digital Scanning of Analog Images, Comparative Characteristics of Digital and Analog Imagery, Spectral Sensitivity, Band Combinations: Optical Imagery, Coverage by Multiple Photographs, Photogrammetry, Sources of Aerial Photography.

Unit III : Digital Imagery
Introduction, Electronic Imagery, Spectral Sensitivity, Digital Data, Data Formats, Band Combinations: Multispectral Imagery, Image Enhancement, Image Display, Image Processing Software.

Unit IV: Land Observation Satellites
Satellite Remote Sensing, Landsat Origins, Satellite Orbits, The Landsat System, Multispectral Scanner Subsystem, Landsat Thematic Mapper, Administration of the Landsat Program, Current Satellite Systems, Data Archives, Active Microwave, Geometry of the Radar Image, Wavelength, Penetration of the Radar Signal, Polarization, Look Direction and Look Angle, Real Aperture Systems, Synthetic Aperture Systems, Interpreting Brightness Values.

Unit V : Thermal Imagery
Introduction, Thermal Detectors, Thermal Radiometry, MicrowaveRadiometers, Thermal Scanners, Thermal Properties of Objects, Geometry of Thermal Images, The Thermal Image and Its Interpretation, Landsat Multispectral Scanner and Thematic Mapper

Thermal Data, Image Resolution, Target Variables, System Variables, Measurement of Resolution, Mixed Pixels, Spatial and Radiometric Resolution: Simple Examples

Unit VI : Hyperspectral Remote Sensing

Introduction, Spectroscopy, Hyperspectral Remote Sensing, The Airborne Visible/Infrared Imaging Spectrometer, The Image Cube, Spectral Libraries, Spectral Matching, Spectral Mixing Analysis, Spectral Angle Mapping, Change Detection: Introduction, Bitemporal Spectral Change Detection Techniques, Multitemporal Spectral Change Detection.

COM1000904: KNOWLEDGE REPRESENTATION AND REASONING

1. **Learning Outcomes**: At the end of the course, students will be able to:
    - (a) Understand the concept of knowledge what is knowledge representation
    - (b) Understand Symbolic and Propositional Logic
    - (c) Learn how Propositional Logic is extended to Predicate Logic
    - (d) Understand First order proof procedures.
    - (e) Learn Logic programming using Prolog.

2. Prerequisite: Mathematical Foundation of Computer Science / Discrete Mathematics or equivalent course.

3. Semester: 10

4. Course type: Elective

5. Course level: 500

6. Theory credit: 3

7. Practical credit: 1

8. Number of required hours:

a) Theory:35 hrs (35 classes)

b) Practical: 30 hrs (15 classes)

c) Non- Contact: 10

9. Reference books:

a) Ronald J. Brachman, Hector J. Levesque: Knowledge Representation and Reasoning, Morgan Kaufmann, 2004.

b) NPTEL

10. Course Outcomes

CO-01	Students will understand the foundations of KRR
CO-02	Learn the trade-off between representation and reasoning
CO-03	Understand which knowledge-based techniques are appropriate for which tasks
CO-04	Students can apply KRR systems to their research and challenging problems
CO-05	Will be able to write a research paper related to knowledge representation

11. Detailed Syllabus

UNIT 1: KEY CONCEPTS

The key concept of knowledge, representation and reasoning. The role of Logic in knowledge representation. History and Philosophy of knowledge and representation.

UNIT 2: PROPOSITIONAL LOGIC

Symbolic Logic: Truth, Logic and Provability. Propositional Logic as a simple knowledge representation language; Syntax and Semantics.

Proof Systems - Natural Deduction, Tableau Method, Resolution Method. Propositional Consequences.

UNIT 3: FIRST ORDER LOGIC

Representing Knowledge in First Order Predicate Logic. Syntax and Semantics. Universal Instantiation. Logical consequence. First order proof procedures: Tableaux and Resolution. The Unification Algorithm. Forward and Backward Chaining. The Resolution Refutation Method.

Reasoning with Horn Clauses, SLD resolution. Logic Programming and Programming in Prolog.

Limitations of Propositional and First Order Predicate Logic.

UNIT 4: FRAGMENTS OF FIRST ORDER LOGIC

Description Logics as Knowledge Representation Languages, Reasoning in Description Logics

Light weight description logics. Horn Fragments of First Order Logic. Rule-based Knowledge Representation and Reasoning.

UNIT 5: RULE BASED SYSTEMS.

The OPS5 Language. The Rete Algorithm.

Representation in First Order Logic. Conceptual Dependency.

Practicals:

1. Simple programs in Prolog.

2. Programs to demonstrate free and bound variables.

3. Programs to demonstrate goals, compound goals and backtracking.

4. Programs to demonstrate using rules to solve problems

5. Implementation of Tableaux and Resolution methods in Propositional Logic.

6. Implementation of Tableaux and Resolution methods in First Order Logic.

INF0900104:ARTIFICIAL NEURAL NETWORK

1. **Learning Outcomes:** At the end of the course, students will be able to:
   (a) Understand linear and non-linear classification.
   (b) Understand supervised and unsupervised learning.
   (c) Apply ANN algorithm for classification

2. **Prerequisites:** Fundamentals of Python programming

3. **Semester:** 9

4. **Course type:** Elective

5. **Course level:** 500

6. **Theory credit:** 3

7. **Practical credit:** 1

8. **Number of required hours:**

   a) Theory:35 hrs (35 classes)
   b) Practical: 30 hrs (15 classes)
   c) Non Contact: 10

9. **Reference books:**
   a) Fundamental of neural networks, LaureneFausett, Pearson
   b) Elements of Artificial Neural Network, 2ed, Kishan Mehrotra, penram International Publishing (India) Pvt. Ltd.
   c) Principles of Soft Computing, SN Sivanandam, SN Deepa, Wiley India
   d) Neural networks and deep learning ,Charu C. Aggarwal, Springer
   e) NPTEL

10. **Course Outcomes:**

CO-01	Analyze linear and non-linear classification etc.
CO-02	Understand role of activation function, training, testing.
CO-03	Understand different components of artificial neural network.
CO-04	Understand unsupervised learning.
CO-05	Apply supervised and unsupervised learning algorithm.

11. Detailed Syllabus:

A. Theory

Unit I : Introduction to ANN 7 Lectures

Introduction to learning, intelligence, and machine learning, applications of ANN, history of neural network, difference between machine learning and deep learning, challenges of machine learning techniques, Biological neuron, artificial neuron, Typical architecture(Single layer net, layered network, fully connected networks, acyclic network, feed forward networks) difference between discriminative and generative learning, common activation function(identity function, binary step function, ramp function,binary sigmoid, bipolar sigmoid), bias .

Unit II: Pattern classification 8 Lectures

Introduction to classification, clustering, pattern association, minimum distance classifier, Euclidean distance, mohalanobis distance, nearest neighbor rule, K-nearest neighbor rule, linear classifier – two class problem, decision boundary between two different classes, design of separating plane, gradient descent algorithm, multi class classification using SVM. Optimizing loss function: Data loss, regularization loss, quadratic loss function, disadvantages of quadratic loss function, cross entropy loss, linear & logistic regression, softmax classifier.

Unit III: Neural Network 7 Lectures

Neural network, input layer, hidden layer, output layer, weight vector, design of neural network using AND function, OR function, XOR function, single layer network without nonlinearity,single layer network with nonlinearity, multilayer perceptron, perceptron training algorithm, choice of learning rate, feed forward neural network, error function: sum of squared, cross entropy, weight updating rule.

Unit IV: Backpropagation Neural Network 8 Lectures

Introduction, architecture, back propagation learning (single layer multiple output, multilayer multiple output), back propagation learning: hidden layer. Some selected applications. Evaluation of networks(quality of results, generalizability, regularization), setting the parametervalues(initialization of weights, frequency of weight updates, choice of learning rate, momentum, number of hidden layers and nodes, number of samples).

Unit V: Unsupervised learning 5 Lectures

Introduction to unsupervised learning, hamming networks, maxnet, simple competitive learning, k-means clustering algorithm, counterpropagation network, Topologically Organized Networks(SOM).

INF900204: SOFT COMPUTING

1. **Learning Outcomes**
   - Learn about advanced forms of set theories including fuzzy and rough.
   - Understand genetic algorithms and their applications.
   - Learn different classification and clustering algorithms based on Rough and Fuzzy sets.

2. **Course Outcomes**

   - CO1: Recognize the difference between Crisp, Fuzzy and Rough sets.
   - CO2: Infer classification and clustering methods based on rough and fuzzy sets.
   - CO3: Relate Fuzzy to Rough set theory.
   - CO4: Prioritize among fuzzy set theory needs and limitations.
   - CO5: Distill different methods of genetic algorithms.

3. **Prerequisites**: Discrete Mathematics or equivalent course
4. **Semester**: 9
5. **Course Type**: Elective
6. **Course Level**: 500-599
7. **Theory Credit:** 4
8. **Practical Credit:** 0
9. **No of Hours:** 60
10. **List of books:**
    1. S.N. Sivanandam and S.N. Deepa, Principles of Soft Computing, Wiley India Edition.
    2. J. Valente de Oliveira, W. Pedrycz, Advances in Fuzzy Clustering and its Applications, John Wiley & Sons, 2007.

11. **Content of syllabus:**

UNIT 1:Fuzzy Sets: Classical(crisp) sets and operations on sets. Fuzzy Sets. Notion of uncertainty of membership in a fuzzy set. Fuzzy set operations and their properties. Classical Relations and Fuzzy Relations. Features of the membership functions, fuzzification, methods of membership value assignments – Intuition, Inference, Rank Ordering and Angular Fuzzy sets. Defuzzification: Introduction, Alpha-Cuts. Defuzzification Methods – Max-membership Principle, Centroid method, Mean-max Membership, Centre of Sums, Centre of Largest Area.

UNIT 2: Rough Sets: Information Systems, Indiscernibility relation, lower and upper approximations; negative and boundary regions of rough sets. Independence of attributes. Core and Reducts of attributes and attribute values. Decision Systems - Dispensable and Indispensable Attributes; Reducts and Cores.

UNIT 3: Fuzzy Clustering: Limitations of hard partitioning and need for fuzzy clustering, Fuzzy c-means (FCM) algorithm.

UNIT 4: Rough Set Based Methods: Information granulation using rough sets, decision rules in rough set models, classification, and clustering methods based on rough sets.

UNIT 5: Genetic Algorithm: What areGenetic Algorithms? Basic terms and terminologies in GA's. Search space. Encoding and Reproduction cycle. Genetic operators – mutation, crossover and selection. Fitness function. Selection methods - Roulette wheel, Stochastic universal sampling, Binary tournament. Elitism. The general Genetic Algorithm with Flowchart. k-means clustering using GA.

INF0900304: BLOCKCHAIN TECHNOLOGY

1. **Learning Outcome:**
   - After completion of this course, students will be able to
   - Understand the basic concepts of blockchain
   - Understand how a blockchain based application works
   - Adopt the concept of blockchain in distributed application development
2. **Course outcomes:**

   After completion of this course, students will be able to
   - CO1: Identify the basic components of a blockchain application
   - CO2: Analyze the working of blockchain based applications
   - CO3: Explain the power and limitations of blockchain based application
   - CO4: Understand the basics of designing DAPP in ethereum platform
3. **Prerequisite:** NIL
4. **Semester:** 9
5. **Course Type:** Elective
6. **Course Level:** 500
7. **Theory Credit:** 4
8. **Practical Credit:** 0
9. **Number of required hours:**
   a) Theory: 60 hrs
   b) Practical: 0 hrs
   c) Non-Contact: 5 hrs
10. **List of reference books:**
    a) Daniel Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress
    b) Mastering Ethereum: Building smart contracts and DAPPS
11. **Detailed Syllabus:**

UNIT 1: Introduction (15 lectures)

Two software architectures: centralized and distributed, advantages of distributed systems, disadvantages of distributed system, distributed peer to peer systems, The Potential of Peer-to-Peer Systems, The Definition of a Peer-to-Peer System, Architecture of Peer-to-Peer Systems, The Link Between Peer-to-Peer Systems and the blockchain, The Potential of the Blockchain, Trust and Integrity in Peer-to-Peer Systems, Integrity Threats in Peer-to-Peer Systems, The Core Problem to Be Solved by the Blockchain, blockchain as data structure, blockchain as algorithm, blockchain as a suite of technologies, blockchain as an umbrella term for purely distributed peer-to-peer systems with a common application area, provisional definition of blockchain, Ownership and Witnesses, Foundations of Ownership, Identification, Authentication and Authorization, purposes and properties of a ledger, ownership and the blockchain, the double spending problem

UNIT 2: How the Blockchain Works (20 lectures)

Documenting ownership with the blockchain: Describing the transfer of ownership, Maintaining the history of transfers, Importance of Ordering, Integrity of the Transaction History, hash functions, properties of cryptographic hash function, SHA-256, patterns of hashing data, Comparing Data, Detecting Changes in Data, Referring to Data in a Change-Sensitive Manner, Storing Data in a Change-Sensitive Manner: chain and tree, markle tree, Causing Time-Consuming Computations, hash puzzles, The Difficulty Level, Cryptography, Symmetric Cryptography, Asymmetric Cryptography, Creating and Distributing the Keys in asymmetric cryptography, Using the Keys, Asymmetric Cryptography in the Blockchain, Digital Signature, Creating a Signature, Verifying Data by Using the Signature, Identifying Fraud by Using the Signature, Digital Signature in blockchain, RSA algorithm, Storing Transactions in the Blockchain Data Structure, Adding New Transactions, Detecting Changes, The Costs of Manipulating the Blockchain Data Structure, Distributing the Data Store Among Peers, purposes of communication among nodes: Keeping existing connections alive, Establishing new connections, Distributing new information, building blocks for Verifying and Adding Transactions: Validation rules, Reward, Punishment, Competition, Peer control, how the process of verification and addition of new transaction works, Consensus algorithms, PoW, PoS, PoB, PBFT, dealing with dishonest behavior, The Role of Fees Within the Blockchain, definition and characteristics of blockchain

UNIT 3: Limitations, types and Applications of Blockchain (10 lectures)

Technical limitations of blockchain, Non Technical limitations of blockchain, Overcoming the limitations, Conflicting goals of blockchain, roots of the conflicts, solving the conflicts, Four versions of blockchain, Consequences of restricting access, revisiting the characteristics of blockchain, generic application patterns, analyzing blockchain applications

UNIT 4: Ethereum and DAPP (15 lectures)

What is ethereum, Bitcoin, Ethereumvsbitcoin, Ethereum: A General-Purpose Blockchain, components of ethereum, The Structure of a Transaction, The Transaction Nonce, Transaction Gas Transaction Recipient, Transaction Value and Data, Special Transaction: Contract Creation, Digital Signatures, The Signature Prefix Value (v) and Public Key Recovery, Separating Signing and Transmission (Offline Signing), Transaction Propagation, Recording on the Blockchain, Multiple-Signature (Multisig) Transactions, What Is a Smart Contract?, Life Cycle of a Smart Contract, Introduction to Ethereum High-Level Languages, Building a Smart Contract with Solidity, The Ethereum Contract ABI, Programming with Solidity: Data Types, Predefined Global Variables and Functions, Contract Definition, Functions, Contract Constructor and selfdestruct

INF0900404:DIGITAL IMAGE PROCESSING

1. **Learning Outcomes:** At the end of the course, students will be able to learn:
   - concept of digital image
   - Image sampling and quantization
   - Image processing in spatial and frequency domain
   - Image segmentation, binary and colour image processing
   - Image compression

2. Prerequisites: Fundamentalsof Python Programming

3. Semester: 9

4. Course type: Elective

5. Course level: 500

6**. Theory credit:** 3

7**. Practical credit**: 1

8. Number of required hours:

a) Theory:35 hrs (35 classes)

b) Practical: 30 hrs (15 classes)

c) Non Contact: 10

9. List of Reference books:

a) R. C. Gonzalez & R. E. Woods - Digital Image Processing, Addison Wesley, 1993.
b) K. Jain - Fundamentals of Digital Image Processing, PHI
c) K. R. Castleman - Digital Image Processing, PHI 1996
d) W. K. Pratt - Digital Image Processing, John Wiley Interscience, 1991
e) Sanjay Sharma – Digital Image Processing, SK Kataria& Sons
f) NPTEL, IITKGP

10. Course Outcomes:

CO-01	Define the scope of digital image processing
CO-02	Understand image acquisition techniques
CO-03	Understand digital image and image segmentation
CO-04	Apply binary and colour image processing
CO-05	Understand image compression

11. Detailed Syllabus:

A. Theory

Unit I : Digital Image processing

Introduction, application of DIP, Image digitization, sampling, quantitation, display, signal reconstruction from samples, convolution concept, signal reconstruction from image, quantizer design, relationship between pixels, basic transform

Unit II : Image transformation

Introduction, Fourier transform, 1D and 2D Fourier transform, Fourier transform in continuous and discrete domain, properties of Fourier transform, FFT, KL transform, Discrete cosine transform, Walsh transform, Handmard transform, Other transforms: Other separable image transforms
and their algorithms.

Unit III : Image enhancement

Enhancement in spital domain, point processing techniques, mask processing, contrast stretching operations, histogram equalization, Spatial Filtering, Frequency Domain Filtering

Unit IV : Image degradation

Frequency domain processing, Image restoration techniques, estimation of degradation model and restoration techniques, image registration, Inverse Filtering, Wiener Filtering.

Unit V :Colour Image processing

Colour fundamentals, different colour models, RGB, CMY, HIS etc.

Unit VI : Image segmentation

Introduction, different approaches for image segmentation, Edge detection, Line detection, Image segmentation: Global processing(Hough Transform), region-based segmentation operation, Thresholding techniques, region splitting and merging

Unit VII : Binary image processing

Binarisation, arithmetic and logical operations, Dilation, Erosion, opening and closing operations, Some basic morphological operations: Boundary extraction, Region filling, Hit-or-Miss transformation, Thinning and ThickeningDistance Transform

Unit VIII : Image compression

Introduction, Types of data redundancies, Image compression model, Lossy Compression, Loss-less compression, Run-length, and Huffman Coding, Transform Coding, LZW compression, Image Compression Standards.

B. Practical

f. Reading and display digital image
g. Image enhancement using spital domain techniques
h. Image segmentation using mask processing and binary image processing techniques
i. Apply image compression technique
j. Drawing histogram for binary and colour image

INF0900504: ADVANCED WEB DEVELOPMENT TECHNIQUE

1. Learning Outcome:

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

- Design basic well-structured web page using HTML and CSS
- Develop the ability to implement interactive elements and dynamic content using basic JavaScript
- Develop a foundational understanding using frameworks

2. COURSE OUTCOMES:

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

CO1: Apply layouts to basic web pages.
CO2: Associate with different scripting languages.
CO3: Explain advance concepts of CSS and JavaScript.
CO4: Distinguish and use different web frameworks for full stack web sites.
CO5: Develop complex and complete web applications.

3. Prerequisite: NIL

4. Semester: 9

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 3

8. Practical Credit: 1

9. Number of required hours:

a) Theory: 45 hrs

b) Practical: 30 hrs

c) Non Contact: 5 hrs

10. List of reference books:

(a) Marcotte, E. (2017). Responsive web design: A book apart. O'Reilly Media.
(b) Duckett, J. (2011). HTML and CSS: Design and Build Websites. John Wiley & Sons.
(c) Robbins, J. N. (2018). Learning Web Design: A Beginner's Guide. O'Reilly Media.
(d) Nixon, R. (2014). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (4th ed.). O'Reilly Media.
(e) Duckett, J. (2014). JavaScript and JQuery: Interactive Front-End Web Development. John Wiley & Sons.
(f) Haverbeke, M. (2018). Eloquent JavaScript. No Starch Press.
(g) React documentation and tutorial: https://legacy.reactjs.org/tutorial/
(h) Angular documentation and tutorial: https://angular.io/tutorial/
(i) Laravel documentation: https://laravel.com/docs/

11. Detailed Syllabus:

Unit 1: Advanced Topics in Web Designing (5 Lectures)

Anatomy of a Web page. Fluid vs Fixed page layout. Fundamentals of responsive design and grid structures. Page design principles. Understanding Rule of thirds in webpage design. Web typography fundamentals.

Unit 2: Advanced HTML and CSS (10 Lectures)

Review of basic HTML tags and Semantic HTML5 elements. Applying proper semantic markup for improved SEO - XML. Multimedia integration. Embedding images with different attributes. Adding video and audio. Meta information and Document Structure – metadata, viewport settings.

Review of CSS fundamentals. Specificity and the cascade. Psudo-classes and pseudo-elements. CSS Box sizing. Gradient and Transparent backgrounds. CSS Typography – Line height and letter spacing. Web-safe fonts. CSS Layout. Display property – inline, block, inline-block, none. Positioning – static, relative, absolute, fixed. Floats and clear property. Box alignment – flexbox and grid layout.Responsive Web Design – Media queries and breakpoints. Fluid layouts. Brief concept of CSS preprocessors – Sass, Less. Brief concept of CSS frameworks – Bootstrap, Tailwind.

Unit 3: Advanced JavaScript (12 Lectures)

Review of JavaScript concepts. Functions in JavaScript. Lexical Environment. Arrays and Array manipulation in JavaScript.. JavaScript Events and Event Handling – Event propagation and event delegation. Implementing interactivity with user actions. Introduction to JavaScript APIs. Callback functions and event loop. Promise chain. Asynchronous function with async/wait. DOM manipulation and event handling with jQuery. Overview of AJAX. Brief concept of XMLHttpRequest object.

Unit 4: Working with Web Frameworks (12 Lectures)

Overview of working with frameworks. Introduction to JavaScript frameworks. React basics – Introduction to JSX and its features. Rendering elements. React Components and Properties. Handling events in React. Overview of React Conditionals, Lists and Keys.

Angular basics – use of TypeScript for Angular. Angular Components – component lifecycle, view encapsulation, interactions, styles, component projection – *ng-content.* Overview of Angular templates.

Laravel basics – Introduction to Laravel structure, PHP Artisan. Using PHP and Blade for frontend. Using Blade directives. LearningLaravel Object Relational Mapper (ORM) – Eloquent. Overview of Routing.

Unit 5: Advanced Web Development Techniques (6 Lectures)

Overview of design patterns – MVC, MVP and MVVM. Comparison of design patterns and their use cases. Overview of Server-Side Includes (SSI). Brief concepts of Web APIs and data integration. Version Control Systems. Brief overview of Continuous Integration and Deployment. Overview of Web security and SSL/TLS. Web analytics and monitoring.

Practical Assignments:

1. Develop a responsive web page using HTML and CSS with a fluid layout that adjusts based on screen size.
2. Implement the Rule of Thirds in webpage design for a photography portfolio website using advanced CSS techniques.
3. Create a web page layout using CSS Grid or Flexbox, showcasing different sections for content, navigation, and footer.
4. Optimize a web page's typography by using web-safe fonts, adjusting line height, and adding letter spacing for better readability.
5. Embed multimedia elements like images, videos, and audio files into a webpage with appropriate attributes and accessibility considerations.
6. Design and code a webpage using Semantic HTML5 elements, ensuring proper use of tags for improved SEO.
7. Build a form with HTML5 and apply CSS styling for form elements, including validation and submission handling using JavaScript.
8. Develop a responsive navigation menu using CSS media queries and breakpoints for different screen sizes.
9. Use JavaScript to manipulate the DOM, create interactive elements, and handle user events like clicks and form submissions.
10. Implement AJAX functionality to fetch and display dynamic content from an external API on a webpage.
11. Create a simple JavaScript application using arrays and array manipulation techniques like filtering and sorting.
12. Design a web page layout using CSS Box Model, applying box-sizing, margins, padding, and borders effectively.
13. Explore and implement CSS gradients and transparent backgrounds for visual enhancements on a webpage.
14. Build a responsive web design using CSS frameworks like Bootstrap or Tailwind CSS, showcasing responsive grid systems and utility classes.
15. Develop a small JavaScript application that uses callback functions, promises, and async/await for asynchronous operations.
16. Create interactive elements on a webpage using jQuery for DOM manipulation and event handling.
17. Build a simple web application using a JavaScript framework like React or Angular, showcasing component-based architecture.
18. Explore and compare different design patterns (MVC, MVP, MVVM) and their application in web development scenarios.
19. Demonstrate the use of Server-Side Includes (SSI) in a web project for code reuse and modularity.

20. Implement version control using Git for a web development project, including branching, merging, and collaboration workflows.

INF0900604:MULTIMEDIA AND GRAPHIC DESIGN

Learning Outcomes:

- Acquire skills to create interesting and interactive components for multimedia.
- Develop the capacities to design, assess, enact with creative projects
- Develop art application, aesthetic judgment, and to increase visual perception and critical thinking skills

Course Outcomes:

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

CO1: Identifythe basic principles of graphic design.

CO2: Apply different graphic designing tools.

CO3: Explain file organization of different multimedia elements

CO4: Analyze the linking inputs of interconnected multimedia systems

CO5: Create communication solutions that address audiences and contexts, by recognizing the human factors that determine design decisions.

3. Prerequisites: Computer Graphics

4. Semester: 9

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 2

8. Practical Credit: 2

9. No of Hours:

a) Theory: 30hrs

b) Practical: 30 hrs

c) Non-Contact: 10hrs

10. List of Books:

- Erlhoff, M., & Marshall, T. (n.d.). *Design Dictionary: Perspectives on DesignTerminology*. Walter de Gruyter.
- Hannah, G. F. (2013). *Elements of design: Rowena Reed Kostellow and the Structure of Visual Relationships*. Princeton Architectural Press.
- Aspelund, K. (2022). *The design process: - with STUDIO*. Bloomsbury Publishing USA.
- Bringhurst, R. (2004). *The elements of typographic style*. Hartley & Marks.
- Weixel, Fulton, Barksdale.Morse, “Multimedia Basics”, Easwar Press 2004.
- Andleigh PK and Thakrar K, “Multimedia Systems”, Addison Wesley Longman, 1999.

- Fred Halsall, “Multimedia Communications”, Addison Wesley, 2000.

11. Contents of Syllabus:

UNIT 1:Introduction to Graphics 02

Graphics Overview: Raster graphics, Vector graphics,Tools of graphics, uses & types of graphics, Meaning and definition of graphics design, Elementsand principles of graphic design, history and evolution of Design.

Practicals: 04

1. Create raster graphics (e.g., a digital painting, photo manipulation) (using Adobe Photoshop or GIMP or a similar software).
2. Design vector graphics (e.g., logo, illustration) using (Adobe Illustrator or Inkscape), focusing on scalable elements.

UNIT 2: Introduction to Design Methodology & Process 06

Understanding the historical perspective of design. Design Basics- Elements and principles of Design, Layout basics, Alignment and hierarchy of order, Design Methodology and process. Introduction to basic Design theories in the context of design. Need finding, Affinity mapping.Mind-mapping, idea development and solution exploration.

Practicals: **03**

1. Analyze a historical design example and discuss its impact on contemporary design.
2. Create a design layout (e.g., website, mobile app)using the principles of alignment and hierarchy.
3. Conduct a need-finding exercise(e.g., redesigning a local business's branding), create affinity maps, and develop design ideas based on the findings.

UNIT 3: Drawing Fundamentals for Graphic Representation 05

Drawing fundamentals , materials and processes. Basics of perspective, scale, light and Composition Understanding tools and techniques for graphic representation. Color Theory: Color models, color palettes, perception of color. Pictorial Composition. Development of Character Design: Anatomy & Proportions, Body types, Character line-ups.

Practicals: **05**

1. Create a series of character sketches exploring different body types and expressions.
2. Experiment with light and shadow to create a realistic drawing of a still life composition.
3. Peer review of an artwork of a famous real person.

UNIT 4: Typography Design 04

Evolution and development of typography. Types of Letterforms its Anatomy, structure and construction. Fundamentals of typographic Design. Type perception and emotion its use in Design. Layout and composition in communication design. Designing with Type- Form, function and Communication. Static and moving types.

Practical: **03**

1. Create typographic logos for fictional brands focusing on visual impact.
2. Design typographic posters using expressive typography to convey a message.

UNIT 4: Introduction to Multimedia **05**

Multimedia Elements, Multimedia Applications, Multimedia System Architecture, Evolving

Technologies for Multimedia Systems, Multimedia Databases; Types of Compression, BinaryImage Compression Schemes, Color, gray scale, still-video image compression, video Imagecompression, audio compression; Data and File format standards- RTF, TIFF, RIFF, MIDI,JPEG, AVI, JPEG.

Practical: **03**

1. Create a story board using multimedia elements (e.g images, audio clips, video snippets) for a thematic project (like promoting sustainability, saving polar bears, etc)

UNIT 5: Multimedia Systems and Security **08**

Key Technology Issues, Pen Input, Video and Image Display Systems, Print Output, Technologies, Image Scanners, Digital Voice and Audio, Video Images and Animation, FullMotion Video, HierarchicalStorage Management, Types of Multimedia systems, Virtual Reality Design, Components of Multimedia system,Digital Rights Management Systems, Technical Trends, Multimedia encryption, Digital Watermarking, Security Attacks: Multimedia Authentication, Pattern, Speaker and Behavior Recognition, Speaker Recognition, Face Recognition.

Practicals: **03**

1. Create a digital watermarking system for multimedia content protection.
2. Implement multimedia encryption techniques to secure sensitive data in a project.

12. List of Final Practicals **10**

1. Design a digital portfolio showcasing all work.
2. Design a brand identity package for a fictional company, including a logo, business cards, letterheads, and other branded materials.
3. Create a series of posters for a social awareness campaign, focusing on issues like environmental conservation, mental health awareness, or community outreach.
4. Develop a magazine layout for a specific genre (e.g., fashion, travel, technology) including cover design, article layouts, and visual hierarchy.
5. Design a set of packaging materials for a new product, considering factors like branding, usability, and aesthetic appeal.

6. Create a series of infographics to visually communicate complex data or statistics on a relevant topic, such as climate change, global health, or technological advancements.
7. Develop a set of digital illustrations or vector graphics for a children's book, educational materials, or storytelling project.
8. Design a set of promotional materials for a cultural event, such as posters, flyers, banners, and social media graphics.
9. Create a series of typographic posters exploring different font styles, typography techniques, and visual communication.
10. Create a multimedia portfolio to showcase various multimedia projects.
11. Develop a multimedia application with interactive features like audio playback and image slideshows.
12. Design a promotional campaign for a social cause using multimedia elements such as videos, infographics, and interactive content.
13. Create an interactive educational module for teaching a specific topic using multimedia components like videos, quizzes, and interactive animations.
14. Develop a multimedia-rich website for a fictional company or organization, incorporating multimedia elements like images, videos, audio, and interactive forms.

INF0900704: ETHICS IN INFORMATION TECHNOLOGY

1. Learning Outcome:

After completion of this course, students will be able to

- Understand the concept of computer ethics
- Implement ethical practices in application development

2. Course outcomes:

After completion of this course, students will be able to

CO1: Understand the concept of ethics

CO2: Adopt ethical practices in computer networks

CO3: Understand and protect intellectual properties

CO4: Adopt information privacy preserving practices

3. Prerequisite: NIL

4. Semester: 9

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 4

8. Practical Credit: 0

9. Number of required hours:

a) Theory: 60 hrs

b) Practical: 0 hrs

c) Non-Contact: 5 hrs

10. List of reference books:

a. Michael J. Quinn, Ethics for the Information Age, Pearson
b. George W. Reynolds, Ethics in Information Technology, Course Technology Inc

11. Detailed Syllabus:

UNIT I. INTRODUCTION (17 lectures)

Introduction to Ethics, Overview of Ethical Theories, Subjective Relativism, The case for Subjective Relativism, The case against Subjective Relativism, Cultural Relativism, The case for Cultural Relativism, The case against Cultural Relativism, Divine Command Theory, The case for Divine Command Theory, The case against Divine Command Theory, Ethical Egoism, The case for Ethical Egoism, The case against Ethical Egoism, Kantianism, The case for Kantianism, The case against Kantianism, Act Utilitarianism, Principle of Utility, Evaluating a Scenario Using Act Utilitarianism, The Case for Act Utilitarianism, The Case against Act Utilitarianism, Rule Utilitarianism, Basis of Rule Utilitarianism, Evaluating a Scenario Using Rule Utilitarianism , The Case for Rule Utilitarianism, The Case against Utilitarianism in General, Social Contract Theory, The Social Contract, Rawls's Theory of Justice , Evaluating a Scenario Using Social Contract Theory, The Case for Social Contract Theory, The Case against Social Contract Theory, Virtue Ethics, Virtues and Vices, Making a Decision Using Virtue Ethics, The Case for Virtue Ethics, The Case against Virtue Ethics, Comparing Workable Ethical Theories, Morality of Breaking the Law, Social Contract Theory Perspective , Kantian Perspective, Rule Utilitarian Perspective, Act Utilitarian Perspective

UNIT II. Networked Communication (13 lectures)

Introduction, Spam, The Spam Epidemic, Need for Social-Technical Solutions, Case Study: Ann the Acme Accountant, Internet Interactions, The World Wide Web, The Rise of the App, How We Use the Internet, Text Messaging, Transforming Lives in Developing Countries, Business Promotion, Political Activism, Censorship, Direct Censorship, Self-Censorship, Challenges Posed by the Internet, Government Filtering and Surveillance of Internet Content, Ethical Perspectives on Censorship, Freedom of Expression, Freedom of Expression Not an Absolute Right, FCC v. Pacifica Foundation, Case Study: Kate's Blog, Children and Inappropriate Content, Web Filters, Child Internet Protection, Ethical Evaluations of CIPA, Breaking Trust, Identity Theft , Chat-Room Predators, Ethical Evaluations of Police Sting Operations, False Information, Cyberbullying, Internet Addiction,Contributing Factors

UNIT III. Intellectual Property and Information Privacy (17 lectures)

Introduction, Intellectual Property Rights, Property Rights, Extending the Argument to Intellectual Property, Benefits of Intellectual Property Protection, Limits to Intellectual Property Protection, Protecting Intellectual Property, Trade Secrets, Trademarks and Service Marks, Patents, Copyrights, Fair Use, BitTorrent, Legal Action against the Pirate Bay, Megaupload Shutdown, Legal Music Services on the Internet, Protections for Software, Software Copyrights, Violations of Software Copyrights, Safe Software Development, Software Patents, Section 3(k) of the Indian Patents Act, Open source software, Consequences of Proprietary Software, "Open Source" Definition, Beneficial Consequences of Open-Source Software, Examples of Open-Source Software, The GNU Project and Linux, Impact of Open-Source Software, Critique of the Open-Source Software Movement, Legitimacy of Intellectual Property Protection for Software, Rights-Based Analysis, Utilitarian Analysis, Creative Commons, defining privacy, Harms and Benefits of Privacy, Is There a Natural Right to Privacy?, Privacy and Trust, Case Study: The New Parents, Information Disclosure, Facebook Tags, Enhanced 911 Services, Rewards or Loyalty Programs, Body Scanners, RFID Tags, Implanted Chips, OnStar, Automobile “Black Boxes”, Medical Records, Digital Video Recorders,Cookies and Flash Cookies, Data Mining, Opt-In versus Opt-Out Policies, Examples of Data Mining,Social Network Analysis, Examples of Consumer Backlash, Marketplace: Households, Facebook Beacon, Netflix Prize, Malls Track Shoppers’ Cell Phones, iPhone Apps Uploading Address Books, Instagram’s Proposed Change to Terms of Service

UNIT IV. IT Law in India (13 lectures)

Information Technology Act 2000, Background, Contents, chapters, 2008 Ammendments, Offences and Penalties, Notable Cases of Section 66, Section 66A, Section 69A, Criticisms: Section 66A and restriction of free speech, Petitions challenging constitutionality, Revocation by the Supreme Court, Strict data privacy rules, Section 69 and mandatory decryption, Section 69A and banning of mobile apps, Information Technology Rules 2021, Digital Personal Data Protection Act 2023, background, Overview, Data Protection Board of India, Rights and provisions, Exemptions, Criticisms: Transfer of Personal Data Outside India, Non-applicability to offline personal data, Misplaced objectives, difference in Right to Privacy verdict interpretation, Exemptions to Government, Obligation with Convention on the Rights of the Child

INF1000104: OPTIMIZATION TECHNIQUES

3. **Learning Outcomes**
   - Understand the need for optimization.
   - Learn and apply different optimization techniques.
   - Learn among different algorithms used for optimizations.

Course Outcomes

CO1: Recognize the need for optimization and different techniques available.
CO2: Appraise between different optimization techniques.
CO3: Illustrate constrained and unconstrained optimization techniques.
CO4: Solve Linear Programming problems using different methods.
CO5: Understand the constrained convex optimization problems.

11. **Prerequisites:** Discrete Mathematics or equivalent course in Mathematics
12. **Semester: 10**
13. **Course Type: Elective**
14. **Course Level: 500**
15. **Theory Credit: 4**
16. **Practical Credit: 0**
17. **No of Hours: 60**
18. **List of books:**

1. Gillette, B.G. : Introduction to operations research - A Computer oriented algorithmic approach, McGraw Hill.
2. N.S. Kambo: Mathematical Programming Techniques; January 2008; EWP.
3. K. V. Mital: Optimization Methods, Wiley Eastern; 3 rd Edition.
4. G. Hadley: Linear Programming, Narosa Publications
5. C. H. Papadimitriou and K. Steiglitz: Combinatorial Optimization - Algorithms and Complexity,
Prentice Hall.

12. **Content of syllabus:**

UNIT 1: Unconstrained optimization:

Necessary and sufficient conditions for optima, convex sets, convex functions, optima of convex functions, steepest descent, Newton and quasi Newton methods, conjugate direction methods.

UNIT 2: Constrained optimization:

Linear Programming - Mathematical model, Basis, feasible solutions and basic feasible solutions, Graphical solution method, unboundedness, Simplex method, Revised simplex method, Applications, Duality, Dual simplex method, Primal Dual Algorithms. Complexity of the algorithms studied. Ellipsoid Method, Karmakar's algorithm.

UNIT 3: Special models of linear programming problems:

Transportation and assignment problems, Maxflow and shortest path problems, Ford and Fulkerson algorithm, Djikstra’s algorithm. Integer programming: Introduction, Travelling Salesman Problem (TSP), Branch and Bound techniques.

UNIT 4: Constrained Convex Optimization:

Problem definition, Kuhn-Tucker Conditions and projected gradient methods.

INF1000204: DATA SCIENCE

2. **Learning Objectives:**
- Understand the fundamental concepts and methods of data science.
- Gain proficiency in data manipulation, visualization, and analysis.
- Learn how to apply statistical techniques to extract insights from data.
- Develop skills in machine learning and predictive modeling.
- Learn some advanced techniques used in data science.

3. **Course Outcomes:** Students will be able to-

CO1: Identify the responsibilities required for a data scientist.

CO2: Differentiate between types of data and understand the importance of data cleaning and manipulation.

CO3: Compute statistical summaries and descriptive statistics to understand the underlying patterns in data.

CO4: Illustrate about text mining, deep learning, and optimization techniques.

CO5: Implement linear regression, logistic regression, various classification and clustering algorithms, PCA, LDA

10. **Prerequisite: NIL**
11. **Semester: 9**
12. **Course Type: Elective**
13. **Course Level: 500**
14. **Theory Credit: 4**
15. **Practical Credit: 0**
16. **Number of required hours:**

a) Theory: 60 hrs
b) Practical: NIL
c) Non Contact: 5 hrs

12. List of reference books:

A. "*Python for Data Science Handbook*" by Jake VanderPlas

B. "*Python for Data Analysis*" by Wes McKinney

C. "*Introduction to Statistical Learning*" by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

D. “*Data Science from Scratch*" by Joel Grus

13. Detailed Syllabus:

UNIT I: Introduction to Data Science (04 Lectures)

Overview on data science, Role of data scientists, Introduction to data analysis tools and libraries (e.g., Python, Pandas, Matplotlib).

UNIT II: Data Preprocessing (08 Lectures)

Types of data, Data cleaning and manipulation, Handling missing values, Handling outliers, Feature engineering.

UNIT III: Exploratory Data Analysis (08 Lectures)

Data visualization techniques, Statistical summaries and descriptive statistics, Probability distributions and random variables, Correlation analysis, Hypothesis testing.

UNIT IV: Machine Learning Basics (20 Lectures)

Overview of supervised and unsupervised learning, Linear regression and logistic regression, Classification algorithms- K-Nearest Neighbor, Decision tree, Clustering algorithms- K-Means, PAM, DBSCAN, BIRCH.

UNIT V: Model Evaluation and Validation (04 Lectures)

Cross-validation techniques, Evaluation metrics for classification, Evaluation metrics for clustering, Evaluation metrics for regression, Overfitting and Underfitting.

UNIT VI: Feature Selection and Dimensionality Reduction (08 Lectures)

Techniques for feature selection, Dimensionality reduction- Principal Component Analysis (PCA), LDA, t-SNE.

UNIT VII: Advanced Topics in Data Science (08 Lectures)

Introduction to Text mining- TF, IDF, vector space model, Introduction to ANN- different types of ANN models, Introduction to CNN- different types of CNN models like VGG-16, LSTM, Introduction to optimization techniques like genetic algorithm.

INF1000304: DEEP LEARNING

1. Learning Outcomes: At the end of the course, students will be able to:
   (a) Understand difference between machine learning and deep learning.
   (b) Understand different deep learning algorithms.
   (c) Apply deep learning algorithms for classification.

2. Prerequisites: Introduction to Machine learning and FundamentalsofPythonprogramming

3. Semester: 10

4. Course type: Elective

5. Course level: 500

6. Theory credit: 3

7. Practical credit: 1

8. Number of required hours:

   a) Theory:35 hrs (35 classes)

   b) Practical: 30 hrs (15 classes)

   c) Non Contact: 10

9. Reference books:

   5. Fundamental of neural networks, LaureneFausett, Pearson
   6. Principles of Soft Computing, SN Sivanandam, SN Deepa, Wiley India
   7. Neural networks and deep learning ,Charu C. Aggarwal, Springer
   8. NPTEL

10. Course Outcomes

CO-01	Understand machine learning and deep learning.
CO-02	Define different optimization techniques in deep learning
CO-03	Understand different components of neural network
CO-04	Understand different unsupervised neural network
CO-05	Solve some real time problem using advanced deep learning models

11. Detailed Syllabus:

A. Theory

Unit I : Introduction to deep learning

Introduction to learning, intelligence, and machine learning, applications of deep learning, difference between machine learning and deep learning, challenges of deep learning techniques, difference between discriminative and generative learning, linear time invariant (LTI) system, linear shift invariant(LSI) system, convolution.

Unit II : Optimization

Optimization in machine learning, Stochastic gradient descent, batch, mini batch optimization, advantages and disadventages of each techniques. Underfitting and overfitting, nonlinearity, nonlinear mapping, nonlinear function : threshold, logistic regression, Rectified Linear Unit(ReLU), sigmoid.

Unit III : Neural Network

History of neural network, biological neuron, artificial neuron, bias, neural network, input layer, hidden layer, output layer, weight vector, design of neural network using AND function, OR function, XOR function, single layer network without nonlinearity,single layer network with nonlinearity , multi layerpreceptron, feed forward neural network, error function: sum of squared, cross entropy, weight updation rule, back propagation learning(single layer multiple output, multi layer multiple output), back propagation learning: hidden layer.

Unit VI : Autoencoder

Applications of autoencoder, Unsupervised learning, label and unlabel data, disadvantages of unsupervised learning, neural network as unsupervised learning, structure of autoencoder, input layer, bottleneck layer, output layer, under complete autoencoder, stacked autoencoder, deep autoencoder, introducation to principal component analysis(PCA), relation between autoencoder vs. PCA, training of autoencoder, introduction to sparse autoencoder, denoising autoencoder.

Unit V : Convolutional Neural Network

Convolution operation, 1D and 2D operation in continuous and discrete domain, finite convolution kernel, feature map, padding, stride, typical architecture of CNN, nonlinearity, pooling, fully connected layer, depth of network, 3D convolution, different CNN models: LeNet 5, IMAGENET Large Scale Visual Recognition Challenge(ILSVRC)

INF1000404: DISTRIBUTED SYSTEM

10. Learning Outcome:

- To provide students the concepts of basic architecture and components of distributed systems
- To familiarize the students with the concepts of various distributed algorithms required for achieving mutual exclusion, leader election and agreements.
- To give students the concepts of inter-process communication and distributed file system handing

11. COURSE OUTCOMES:

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

- CO1: Explain the concept and challenges for designing a distributed system.
- CO2: Describe the architecture and different system models of distributed systems.
- CO3: Analyze different synchronization algorithms used in distributed systems.
- CO4: Illustrate different algorithms used for achieving mutual exclusion and agreements in distributed applications.
- CO5: Explain the concept of Inter-process communication techniques in distributed systems.

12. Prerequisites: NIL

13. Semester:10

14. Course Type: Elective

15. Course Level: 500

16. Theory Credit: 4

17. Practical Credit: 0

18. No of Hours:

a) Theory: 60hrs

b) Practical: 0hrs

c) Noncontact: 5hrs

10. List of Books:

a) Tanenbaum & Steen; (2004); Distributed Systems Principles and Paradigms; Pearson Education

b) Coulouris, Dollimore&Kindberg; (2006); Distributed Systems Concepts and Design; Pearson Education

c) Udit Agarwal; (2016); Distributed Systems; S. K. Kataria& Sons

11. Contents of Syllabus:

UNIT-1 Introduction to Distributed Systems (6 hours)

Definition of a distributed system, Characteristics of distributed and centralized systems, examples of distributed systems: *internet, mobile and ubiquitous computing*, Difference between computer network and a distributed system, Challenges in distributed systems: *heterogeneity, openness, security, failure handling, concurrency* and *transparency.*

UNIT-2 System Models (8 hours)

Problems faced by designers of a distributed system, types of system models: *Physical model, architectural model* and *fundamental model.* Architectural model: software layers, software architectures (*client-server, peer-to-peer*), variations of client-server architectures, design requirements for distributed architectures. Fundamental models:definitionof fundamental model, concept of interaction model, factors effecting interaction of processes, variations of interaction model, Failure model, types of failures-*omission failures, arbitrary failures* and *timing failures*, failure masking. Security model-protecting objects, securing processes and their interactions, secure channels, uses of security models.

UNIT-3 Synchronization (12 hours)

Inherent limitations of a distributed system, external and internal clock synchronization, physical and logical clocks, Cristian clock algorithm, Berkeley clock algorithm, Network Time Protocol, Causal order of events, happened before relationship between events, Lamport's logical clocks, limitations of Lamport's clock. Vector clocks, Synchronization of events using vector clocks, Causal Order of messages and Birman-Schiper-Stephension protocol. Definition of local states and global states, consistent, inconsistent and strongly consistent global states, ChandyLamport snapshot global state recording algorithm, need of termination detection, Haung's termination detection algorithm.

UNIT-4 Distributed Mutual Exclusion (8 hours)

Requirements of Mutual Exclusion algorithms, Performance measurement metrics, Classification of mutual exclusion algorithm, Token based algorithms, non-token-based algorithm, Central Server Algorithm, Lamport's timestamp algorithm, Ricart-Agrawala algorithm, Raymond's Tree-based algorithm, Election algorithms- the Bully algorithm.

UNIT-4 Agreement Protocols (6 hours)

Definition of agreement in distributed system, System models and aspects of recognizing the agreement protocols, classification of agreement problems (Byzantine, Consensus, Interactive), Relation among agreement protocols, Solutions to the Byzantine agreement problem: *upper bound of number of faulty processes* and *impossibility results*, Applications of agreement algorithms,

UNIT-5 Distributed Scheduling (4 hours)

Distributed scheduler, issues in distributed load distribution, components of load distribution algorithm, need of load distribution algorithms, classification of load distribution algorithms-*sender-initiated, receiver-initiated* and *symmetricallyinitiated*, performance, task migration.

UNIT-6 Interprocess Communication (8 hours)

Introduction to Inter-process Communications, Communication patterns-*Client-serve communication* vs *Group communication,* Characteristics of interprocess communication: synchronous and asynchronous communication, reliability, message destination. Sockets, UDP datagram communication: issues in datagram communication*,* use of UDP. TCP stream communication: abstraction provided by the API to TCP protocol. External data representation and marshalling, CORBA's common data representation (CDR), remote object reference. Client-server communication, concept of RPC and RMI, Implementation of group communication-IP multicast.

UNIT-7 Distributed File systems (8 hours)

Introduction: characteristics of file systems, distributed file system requirements, Architecture of distributed file system, Desirable features of a good distributed file system, structured vs unstructured files, mutable vs immutable files. Mechanism for building DFS, design issues of a DFS, dependency factors of file accessing models, file caching schemes.

INF1000504:SPEECH PROCESSING

1. Learning Outcomes: At the end of the course, students will be able to:

5. Toprovidetheconceptsofbasicmodelsforspeechproduction and perception
6. Todeveloptimeandfrequencydomaintechniquesforestimatingspeechparameters
7. Tointroducestudentsapredictivetechniqueforspeech recognition.
8. Toprovidestudentstheprocessofspeechrecognition,synthesisand Prosody modeling.

2. Prerequisites: Knowledge of Python programming

3. Semester: 10

4. Course type: Elective

5. Course level: 500

6. Theory credit: 3

7. Practical credit: 1

8. Number of required hours:

a) Theory:35hrs (35 classes)

b) Practical: 30 hrs (15 classes)

c) Non- Contact: 10

9. **Reference books:**

- L. Rabinerand B. Juang,FundamentalsofSpeechRecognition,PrenticeHall,1995,ISBN0-13-015157-2
- L.R.RabinerandR.W.Schafer,DigitalProcessingofSpeechSignals,Prentice-Hall,1978,ISBN0-13- 213603-1.
- J.LFlanagan:SpeechAnalysisSynthesisandPerception-2ndEdition-SprengerVertag,1972.
- NPTEL

10. Course Outcomes

CO-01	Explainbasiccharacteristicsofspeechsignalinrelationtoproductionandhearingofspeechbyhumans.
CO-02	Analyzedifferent time and frequency domainalgorithms ofspeechfeature extraction.
CO-03	Understand speech synthesis
CO-04	Understand trends in developments of automatic speech recognition
CO-05	Explain Test-To-Speech(TTS) system

11. Detailed Syllabus:

A. Theory

UNIT-I: The Speech Signal

The process of speech production and perception in human beings, speech production process, representing speech in time and frequency domain, speech sounds and features: vowels, diphthongs, semivowels, nasal consonants, unvoiced fricative, voice fricative, voiced and unvoiced stops, approaches to automatic speech recognition by machine: acoustic phonetic approach, pattern recognition approach, artificial intelligence approach, source filter model of speech, manners and place of articulation, speech perception, auditory system, human ear

UNIT-II: Speech features

Perception of sound, physical dimension of sound, threshold of hearing, Formant frequency, time domain methods in speech processing, short time energy, short time average magnitude, short time zero crossing rate, short time autocorrelation, Linear predictive coding, segmental and supra segmental features, frequency domain in speech processing: filter bank analysis, short time spectral analysis, cepstral transfer coefficient(CC), mel frequency cepstral coefficients and its variants, fundamental frequency, fundamental frequency characteristics, fundamental frequency extraction methods

UNIT-III: Text To Speech Synthesis

Introduction, steps in text to speech synthesis system, text normalization, text processing, speech synthesis markup language, grapheme to phoneme(G2P), G2P methods, prosody, TTs synthesis models: articulatory, parametric, concatenative, W3C standards, prosody modeling, prosodic structure,

UNIT-IV:**Isolated Speech recognition using machine learning**

Introduction, machine learning, deep learning, linear classification, multi class classification problem, biological neuron, artificial neuron, activation function, loss function, optimizers, designing ANN, speech recognition using any one techniques(SVM, KNN, ANN, CNN, LSTM)

B**. List of Practicals**

(This is a suggestive list only. Questions need not be restricted to this list. The practical are advised to be performed in Linux environment using C programming language.)

1. Record speech signal for 1 sec duration in 8KHz and 16 KHz(Both mono and stero)

2. Write program to find ZCR, LPC, CC, MFCC for speech signal

3 Desing SVM classifier for binary and multiclass classification.

4. Desing ANN for binary and multiclass classification.

5 Design CNN for multiclass classification.

INF1000604:VIDEO EDITING AND ANIMATION

1. **Learning Outcomes:**
   - Introduction to the basic concepts audio and video editing and attain professional skills in them.
   - Get an insight in animation production.
   - Design for various requirements evolved in multimedia production.

2. **Course Outcomes:**

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

CO1: Identify the basic principles of visual effects.

CO2: Apply different editing tools in audio and video.

CO3: Explain among different video and audio editing elements.

CO4: Analyze the different forms of animations and visual effects.

CO5: Create different aminations using various techniques.

3. Prerequisites: Graphic Design and Multimedia

4. Semester: 10

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 2

8. Practical Credit: 2

9. No of Hours:

a) Theory: 30 hrs

b) Practical: 30 hrs

c) Non-Contact: 10 hrs

10. List of Books:

1. Compositing Visual effects: Steve Wright
2. Digital Compositing in Depth: Doug Kelley.
3. Notes on Digital Film Editing: Gerhard Schumm
4. Audio post production for film and Video: Jay Rose
5. Non-linear Editing:Bryce Button
6. The Animator's Survival Kit: Richard Williams

7. Timing for Animation: Harold Whitaker and John Halas

11. Contents of Syllabus:

UNIT 1:HistoryofAnimationandVisual Effect

What is Compositing? Introduction to After Effects Interface Create a new composition, Timeline panels, Adding footage, Resolution, Quality, History of Animation (Theory) – American, Japanese, Indian, and European, AnimationPrinciples – 12 basic principles,Layer Management Selecting - Moving layers, Trim in and out points, Motion blur, Masking Create Masks - Transforming masks, Mask points.

UNIT 2:Audiography

Sounds in nature: Nature of sound waves, Propagation process, Frequency, Amplitude, Wavelength, Pitch, Velocity of travel in different media, Loudness, Pitch, Reverberation, Reflection, Absorption, Diffraction, Refraction, Interference, Delay and Echo.Sound in Recording, Analogous and digital audio technologies., Microscopes, Digital Sound recording: Introduction to sound editing software. How to open a session, Different tools, importing audio files, Introductions to Basic Editing and mixing; Single track Recording, Multi track recording.

<u>Practicals:</u> **10**

1. Create a mini audio documentary or podcast.
2. Record a monologue or narration, apply basic editing techniques like trimming, fading, and adding effects.

UNIT 3: Videography

Video Editing- Need of editing- Linear and Non-linear editing- Stages of Editing process. Factors for good edit-understanding the footage, Matching eye line. Continuity in Cinema- Physical continuity and Emotional Continuity-plot line- Story structure- sound.Time concept – Real time and Filmic time- Cut and editing principles- Jump cut, match cut, cross cut, reverse cut, cut away, cut in.Introducing a Non-Linear Editing Software- Project setting- Introducing interface, Continuity Editing, Match cut, cross cut, jump cut, Dialogue overlapping, L-Cut, J-Cut, Slow motion Fast motion, Synchronizing and mixing Video and Audio

<u>Practicals:</u> **10**

1. Plan and execute a video editing project from raw footage to final output.
2. Practice editing dialogue scenes for smooth transitions.
3. Add visual effects like transitions, overlays, and text animations to enhance the visual appeal of above video.

UNIT 4: Basics of Digital Animation

Understanding of different types of animation AKA production pipeline – Animation Process and style – 2D classical animation – 3D animation – Stop Motion Animation, Animation Equipment– Cells - Light Box– Peg Holes and Peg Bars – Line/Pencil Tests - Field Charts - Rostrum Camera - The Exposure Sheet (X Sheet) – Concepts Of: -Soundtrack, Track Breakdown, Key Frames, In-Betweens, Clean-Up, Animating Special Effects: - Cloth, Sky, Lightening, Rainfall, Snow, Water Drops, Water Ripples, Waves, Smokes, Fire, Explosions Etc.Animation of Four Legged and Two Legged Animals - Normal and Stylized Movements of animals

Practicals: **10**

1. Explore different animation styles such as cartoonish, realistic, and abstract, showcasing your versatility as an animator.
2. Experiment with stop motion techniques using everyday objects or clay models, creating short animated sequences as well as add special effects like water simulations, fire, explosions, and environmental elements.

INF1000704: CYBER SECURITY

1. Learning Outcomes:

- Provide an overview of the field of information security and assurance.
- Develop an understanding of cryptography and to find the vulnerabilities in programs and to overcome them.
- Identify the know the different kinds of security threats and its solution.
- Have an ability to use geospatial technologies to gain a significant advantage in the information technology field.

2. Course Outcomes:

On successful completion of course, learner will be able to:

CO1: Develop and familiarize with various types of cyberattacks, cybercrimes, vulnerabilities and remedies.

CO2: Identify and evaluate the importance of personal data its privacy and security.

CO3: Analyze and evaluate the cyber security risks.

CO4: Evaluate and communicate the human role in security systems.

CO5: Implement preventive measures for self-cyber-protection as well as societal cyber-protection.

3. Prerequisites: Computer Networks, System Administration and Networking

4. Semester: 10

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 4

8. Practical Credit: 0

9. No of Hours:

a) Theory: 60 hrs
b) Practical: 0 hrs
c) Non-Contact: 5 hrs

10. List of Books:

- Donaldson, S., Siegel, S., Williams, C.K., Aslam, A., "Enterprise Cyber security - How to Build a Successful Cyber defense Program against Advanced Threats", Apress, 1st Edition, 2015.
- Nina Godbole, SumitBelapure, "Cyber Security", Willey, 2011.
- Roger Grimes, "Hacking the Hacker", Wiley, 1st Edition, 2017.
- Cyber Law By Bare Act, Govt of India, It Act 2000

11. Contents of Syllabus:

Unit 1: Components of Cyber Security 10

Introduction, Computer Security, Threats, Harm, Vulnerabilities, Controls, Authentication, Access Control and CryptographyCyber security terminologies- Cyberspace, attack, attack vector, attack surface, threat, risk, vulnerability, exploit, exploitation, hacker.,Malware and its types, Web attack: Browser Attacks, Web Attacks Targeting Users, Obtaining User or

Website Data, Email Attacks, Network Vulnerabilities, Network Defense tools, Firewalls and Packet Filters,.

Unit 2: Cyber Law and Cyber-crimes 8

Cyber-crime and legal landscape around the world,Classification of cybercrimes, Cyber-crimes targeting Computer systems and Mobiles- data diddling attacks, spyware, logic bombs, DoS, DDoS, APTs, virus, Trojans, ransomware, data breach., cyber-crime against women and children, financial frauds, social engineering attacks, malware and ransomware attacks, zero day and zero click attacks,Online job fraud, Online sextortion, Debit/ credit card fraud, Online payment fraud, Cyberbullying, website defacement, Cybersquatting, Pharming, Cyber espionage, Crypto jacking, Darknet- illegal trades, drug trafficking, human trafficking., Social Media Scams & Frauds- impersonation, identity theft, job scams, misinformation, fake news, cyber-crime against persons - cyber grooming, child pornography, cyber stalking, Social Engineering attacks, Cyber Police stations, Crime reporting procedure, Case studies.

Unit 3: Data Privacy and Security 6

Defining data, meta-data, big data, nonpersonal data. Data protection, Data privacy and data security, Smartphone Security Cyber-attacks, types of attacks motivation, Wireless security, Authentication and password security, social media- data privacy and security issues.

Unit 4: Hacking Techniques 7

Overview of Hacking, System Hacking, Sniffers, Trojans, Backdoors, Viruses and Worms, Session Hijacking, Social Engineering, Denial of Service, SQL Injection, IDS, Firewalls and Honeypots, Hacking Wireless Networks, Cross site Scripting, Single sign On,Buffer Overflow, password Cracking, Keyloggers and Spyware.

Unit 5: Cryptography 10

Computational Complexity, GCD Computation, Modular Arithmetic, Key Exchange, Public Key Cryptosystem, Factorization, Diffie-Hellman, AES, RSA, Elliptic Curve Cryptosystem, Hash Function: MAC, MD4&MD5, SHA, HMAC, CMAC, Digital Signatures, Ciphers, Data compression-- Loss less &Lossy, Entropy encoding.

Unit 6: E-Commerce, Digital payments and its security 7

Definition of E- Commerce, Main components of E-Commerce, Elements of E-Commerce security, E-Commerce threats, E-Commerce security best practices, Introduction to digital payments, Components of digital payment and stake holders, Modes of digital payments- Banking Cards, Unified Payment Interface (UPI), e-Wallets, Unstructured Supplementary Service Data (USSD), Digital payments related common frauds and preventive measures.

Unit 7: Cyber security of digital devices 7

End Point device and Mobile phone security, Password policy, Security patch management, Data backup, Downloading and management of third-party software, Device security policy, Cyber Security best practices, Significance of host firewall and Ant-virus, Management of host firewall and Anti-virus, Wi-Fi security, Configuration of basic security policy and permissions.

Unit 8: Computer Forensics and Security Analysis 5

Introduction to digital forensic, Understanding storage media and file system, Windows and network forensics, Recovering from Information Loss, Web and email attacks, Packet Analysis & Risk Management, Email Security Analysis, Malware Analysis.

INF1000804: INTERNET OF THINGS

1. Learning Outcome:

After completion of this course, students will be able to

- Understand the basic concepts of IoT
- Identify the building blocks of IoT
- Understand the process of integrating the components of IoT into an integrated system

2. Course outcomes:

After completion of this course, students will be able to

CO1: Identify the basic components of an IoT based system

CO2: Understand various constituent technologies

CO3: Analyse the application of various enablement platforms in IoT based systems

3. Prerequisite: NIL

4. Semester:10

5. Course Type: Elective

6. Course Level: 500

7. Theory Credit: 4

8. Practical Credit: 0

9. Number of required hours:

a) Theory: 60 hrs

b) Practical: 0 hrs

c) Non Contact: 5 hrs

10. List of reference books:

a. Pethuru Raj, Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms and Use Cases, CRC Press

11. Detailed Syllabus:

UNIT 1: Introduction (10 lectures)

Definition of IoT, Why the IoT is Strategically Sound, The Brewing and Blossoming Trends in IT Space, Envisioning the Internet of Things Era, Illustrating the Device-to-Device/ Machine-to-Machine Integration Concept, Explaining the Aspect of Device-to-Cloud (D2C) Integration, The Emergence of the IoT Platform as a Service (PaaS), Digging into the Cloud-to-Cloud (C2C) Integration Paradigm, Describing the Sensor-to-Cloud Integration Concept, Azure IoT Hub Device Management, The Prominent IoT Realization Technologies, The IoT: The Key Application Domains, The Emerging IoTFlavors

UNIT 2: Wireless Technologies (10 lectures)

Introduction to wireless technology in IoT, Architecture for IoT Using Mobile Devices, Energy Harvesting for Power Conservation in the IoT System, Mobile Application

Development Platforms, Mobile Use Cases for IoT, Low Power Wide Area Networking Technologies

UNIT 3: Protocols (10 lectures)

Introduction to IoT protocols, Layered Architecture for IoT, Protocol Architecture of IoT, Infrastructure Protocols, Device or Service Discovery for IoT, Protocols for IoT Service Discovery, Prominent IoT Service Discovery Products Available in the Market

UNIT 4: Integration Technologies (15 lectures)

Introduction to Integration Technologies, Sensor and Actuator Networks, Machine-to-Machine Communication, Service Oriented Device Architecture for Device Integration, Device Profile for Web Services, DPWSim Development Toolkit, The Open Service Gateway Initiative Standard, The REST Paradigm, The Device Connectivity via the OPC Standard, The Device Integration Protocols and Middleware, The Protocol Landscape for IoT

UNIT 5: 5 Enablement Platforms for IoT Applications and Analytics (15 lectures)

The IoT Building Blocks, IoT Application Enablement Platforms, Characterizing IoT or Machine-to-Machine Application Platforms, IoT AEPs—The Architectural Building-Blocks, IoT and M2M Sensor Data Platform by AerCloud, ThingWorxIoT AEP, ORBCOMM IoT Platform, Azure IoT Hub, Amazon Web Service IoT Platform, The AxedaIoT Platform, The IoT Data Analytics Platforms, IBM Watson IoT Platform, ParStreamIoT Analytics Platform, VitriaIoT Analytics Platform, PentahoIoT Analytics Platform, Splunk Software for IoT Data, GuavusIoT Analytics Platform, The IoT Data Virtualization Platforms, The Key Capabilities Data Virtualization Delivers, IoT Data Visualization Platform, The IoT Edge Data Analytics

INF1000904: SOFTWARE TESTING

1. **Learning Outcome:**

- To introduce the basic challenges and concepts Software Testing process.
- To familiarize with structural testing and functional testing approaches.
- To discuss various testing levels and testing tools
- Enable the students to design test cases in different software testing methods.

2. COURSEOUTCOMES:

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

CO1: Explain the challenges and need the of software testing process.

CO2: Describe various methods used in performing Functional testing and Structural testing of a software product.

CO3: Understand different testing levels and testing tools.

CO4: Discuss the process of software verification and design test cases for Object Orientedtesting method.

CO5: Analyze different Software Testing metrics and automated Test Data Generation process

3. **Prerequisites: NIL**
4. **Semester: 10**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **No of Hours:**
   a) **Theory: 60 hrs**
   b) **Practical: 0 hrs**
   c) **Non-Contact: 5 hrs**
10. **List of Books**

(a) Yogesh Singh; (2012); Software Testing; Cambridge University Press

(b) Srinivasan Desikan and Gopalaswamy Ramesh; (2006)'; Software Testing – Principles and Practices, Pearson Education,

(c) Mauro Pezze; (2011); Software Testing and Analysis-Process, Principles and Techniques; John Wiley and Sons.

11. **Contents of the Syllabus**

UNIT-1 Introduction to Software Testing (10 hours)

Some Software Failures: The Explosion of the Ariane 5 Rocket, The Y2K Problem, The USA Star-Wars Program, Failure of London Ambulance System, USS Yorktown Incident , Accounting Software Failures, Testing Process, Definition of importance of Software Testing, Some Terminologies: Program and Software, Verification and Validation, Fault, Error, Bug and Failure, Test, Test Case and Test Suite, Deliverables and Milestones, Alpha, Beta and Acceptance Testing, Quality and Reliability, Testing, Quality Assurance and Quality Control, Static and Dynamic Testing, Testing and Debugging, Limitations of Testing, The V-Shape Software Development Life Cycle model.

UNIT-2 Functional Testing (10 hours)

Introduction to functional testing, Boundary Value Analysis (BVA), Robustness Testing, Worst-case testing, Equivalent Class Testing, Decision Table based testing, Applications of Decision Table based testing, Cause-effect graphing technique, Constraints in Cause-Effect Graph, Essentials of graph Theory: definition of graph, Degree of nodes, Path and independent paths, generation of graph from a program, program graphs, Cyclomatic Complexity.

UNIT-3 Structural Testing (8 hours)

Control Flow testing, Statement Coverage, Branch Coverage, Condition Coverage, Path Coverage, Data Flow testing; Identification of du and dc Paths, generation of test cases, Slice based testing: Guidelines for slicing, creation of Program Slices, generation of test cases, Mutation Testing: Mutation and Mutants, Mutation Score.

UNIT-4 Software Verification (12 hours)

Software Verification Methods: Peer Reviews, Walkthroughs, Inspections, Applications, Software Requirements Specification (SRS) Document Verification, Source Code Reviews: Issues Related to Source Code Reviews, User Documentation Verification, Review Process Issues, Software Project Audit, Creating Test Cases from Requirements and Use Cases, Use Case Diagram and Use Cases, Generation of Test Cases from Use Cases, Guidelines for generating validity checks, Strategies for Data Validity. Software Testing Activities, Levels of Testing: Unit Testing, Integration Testing, System Testing, Acceptance Testing; Debugging: Debugging Process, Debugging Approaches and Tools; Software Testing Tools: Software Testing Tools, Dynamic Software Testing Tools, Process Management Tools, Software Test Plan.

UNIT-5 Object Oriented Testing (6 hours)

Classes and Objects, Inheritance, Messages, Methods, Responsibility, Abstraction, Polymorphism, Encapsulation, Object Oriented Testing: Unit, Levels of Testing; Path Testing: Activity Diagram, Calculation of Cyclomatic Complexity, Generation of Test Cases, State Based Testing: State Machine, State Chart Diagram, State Transition Tables, Generation of Test Cases, Class Testing, Generating Test Cases for Class Testing.

UNIT-6 Performance and Regression Testing (8 hours)

Factors Governing Performance testing, methodology for performance testing: collecting requirements, writing test case, automating performance testing, Executing and Analyzing performance test cases. Performance tuning, Tools for Performance Testing, Definition of Regression testing, Types of Regression Testing, When and How to do Regression testing, performing initial “Smoke” and “Sanity” test, Selecting and Classifying test cases, Mythology for selecting test cases.

UNIT-7 Metrics in Software Testing and Automated Test Data Generation (6 hours)

Software Metrics, Categories of Metrics: Product and Process Metrics for Testing; Object Oriented Metrics: Coupling Metrics, Cohesion Metrics, Inheritance Metric, Size Metrics.

Measures During Testing: Time, Quality of Source Code, Source Code Coverage, Test Case Defect Density, Review Efficiency. Automated Test Data Generation: Definition, Test Adequacy Criteria, Static and Dynamic Test Data Generation, Approaches to Test Data Generation, Random Testing, Symbolic Execution, Dynamic Test Data Generation.

INF1001004: AUGMENTED REALITY

1. **Learning Outcome:**

- Student will be able tounderstand augmented reality concepts
- Student will havebasic understanding of computer vision for augmented reality.
- Students will be able to understand various interaction techniques and interaction styles thatare relevant for augmented reality applications.

2. **COURSEOUTCOMES:**
3. After completion of this course, students will be able to
   CO1: Define augmented reality.
   CO2: Summarize fundamental enabling technology for augmentedreality.
   CO3: Discuss the characteristics that are necessary to understand how tracking and measurement systems in general work.
   CO4: Identify the issues of various augmented reality and virtual realityframeworks.
   CO5: Analyze the underlying architectures of augmentedreality systems.
3. **Prerequisite:** Computer Graphics
4. **Semester: 10**
5. **Course Type: Elective**
6. **Course Level: 500**
7. **Theory Credit: 4**
8. **Practical Credit: 0**
9. **Number of required hours:**
   a) Theory: 60 hrs
   b) Practical: 0 hrs
   c) Non Contact: 5 hrs
10. **List of reference books:**

a. Dieter Schmalstieg& Tobias Höllerer, *Augmented Reality: Principles and Practice*; Pearson Education India.
b. Jerald, Jason, *The VR book: Human-centered design for virtual reality*. Morgan & Claypool., 2015.
c. Forsyth, D. A. & Ponce, J, *Computer Vision: A Modern Approach*, Pearson, 2012.

11. **Detailed Syllabus:**
    UNIT 1: Introduction to Augmented Reality (8 Lectures)

Definition and Scope, A Brief History of Augmented Reality, AR technology and applications, Related Fields, Multimodal Displays, Visual Perception, Requirements and Characteristics, Spatial Display Model Visual Displays

UNIT 2: Tracking, Calibration and Registration, Computer Vision (12 Lectures)
Tracking, Coordinate Systems, Characteristics of Tracking Technology, Stationary Tracking Systems, Mobile Sensors, Optical Tracking, Sensor Fusion, Marker Tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Incremental Tracking, Simultaneous Localization and Mapping, Outdoor Tracking, Camera Calibration, Display Calibration, Registration

UNIT 3: Visual Coherence (8 Lectures)
Registration, Occlusion, Photometric Registration, Common Illumination, Diminished Reality, Camera Simulation, Situated Visualization- Challenges; Visualization Registration; Annotations and Labelling, X-Ray Visualization, Spatial Manipulation, Information Filtering,

UNIT4: Interaction (8 Lectures)
Output Modalities, Input Modalities, Tangible Interfaces, Virtual User Interfaces on Real Surfaces, Augmented Paper, Multi-view Interfaces, Haptic Interaction, Multimodal Interaction, Conversational Agents

UNIT 5: Modelling and Annotation (6 Lectures)
Specifying Geometry- Points;Planes; Volumes, Specifying Appearance, Semi-automatic Reconstruction, Free-Form Modelling, Annotation,

UNIT 6:Authoring&Navigation (10 Lectures)
Requirements of AR Authoring, Elements of Authoring, Stand-Alone Authoring Solutions, Plug-In Approaches, Web Technology, Foundations of Human Navigation, Exploration and Discovery, Route Visualization, Viewpoint Guidance, Multiple Perspectives

UNIT 7:Software Architectures (8 Lectures)
AR Application Requirements, Software Engineering Requirements, Distributed Object Systems, Dataflow, Scene Graphs, Developer Support