Proposed Syllabus (up-to Semester-4)

For

B.Sc. in Computer Science [Major & Minor]

Submitted

To

University of Gour Banga

Under
New Education Policy-2020
[With effect from the Session 2023- 24**]**

Contents

Abbreviations:

MC: Major Core.
DSE: Discipline Specific Elective (Student has to choose one elective subject from each DSE).
MDC: Multidisciplinary Core.
MnC: Minor Core.
AEC: Ability Enhancement Course.
SEC: Skill Enhancement Course.
IAPC: Internship / Apprenticeship / Project / Community Outreach Course.
VAC: Value Addition Course.

Note:

A. *A student with **Computer Science as major** will read the **MC/DSE and SEC** paper for the concerned semester from this syllabus. A student has to choose one paper from respective DSE pool as per the availability of the subject in the concerned institution. Students with major have to choose one subject as minor from the respective pool as per University guidelines. MDC, AEC, IAPC and VAC papers are compulsory, major students has to choose it as per the available guidelines provided by the Institution/ University.*
B. *A student with **Computer Science as minor** will read the **MnC** paper only for the concerned semester from this syllabus.*
C. *A student with **any major (except Computer Science as major)** is allowed to take the **MDC** paper for the **semester – I** from this syllabus.*

Credit Distribution

Sem	MC / DSE	AEC	MnC	MDC	IAPC	SEC	VAC
I	**MC1:** (4) **MC2:** (3+1=4)	(2)	**MnC1:** (3+1=4)	**MDC1:** (3)		**SEC1:** (3)	(2)
II	**MC3:** (3+1=4) **MC4:** (4)	(2)	**MnC2:** (3+1=4)	**MDC2:** (3)	2* (Optional)	**SEC2:** (2+1=3)	Need to choose one(2)
Students on exit shall be awarded Undergraduate Certificate in Computer Science after securing the requisite 44+2*=46 credits in Semester I and II after completion of Summer Internship(2) (optional)*.							
III	**MC5:** (3+1=4) **MC6:** (3+1=4)	(2)	**MnC3:** (3+1=4)	**MDC3:** (3)		**SEC3:** (3)	
IV	**MC7:** (4) **MC8:** (4) **MC9:** (4)	(2)	**MnC4:** (3+1=4)		2**		
Students on exit shall be awarded Undergraduate Diploma in Computer Science after securing the requisite 84 credits in Semester IV after completion of Summer Internship(2) (either in Semester II* or in Semester IV**) (Candidates already pursued IAPC in Semester II, need not to pursue IAPC in Semester IV) .							
V	**MC10:** (4) **MC11:** (4) **DSE-1:** (4) **DSE-2 (4)**		**MnC5:** (4)				Need to choose one(2)
VI	**MC12:** (4) **MC13:** (4) **DSE-3:** (4) **DSE-4:** (4)		**MnC-6:** (4)				
Students on exit shall be awarded Undergraduate Degree in Computer Science after securing the requisite 126 credits in Semester VI.							
VII	**MC14:** (4) **DSE-5: (**4) **DSE-6:** (4)		**MnC-7:** (4)				(4)
VIII	**MC15:** (4) **DSE-7:** (4) **DSE-8:** (4)		**MnC-8:** (4)				(4) + (4)
Students on exit shall be awarded Bachelor of Computer Science Honours with Research / Academic Projects/ Entrepreneurship or Honours with Research in Computer Science (major) with Discipline-2 (minor) after securing the requisite 170 credits on completion of Semester VIII.							
VII	**MC14:** (4) **DSE-5: (**4) **DSE-6:** (4) **DSE-7: (4)**		**MnC-7:** (4)				
VIII	**MC15:** (4) **DSE-8:** (4) **DSE-9:** (4) **DSE-10: (4)**		**MnC-8:** (4)				(4)
Students on exit shall be awarded Bachelor of Computer Science Honours in Computer Science (major) with Discipline-2 (minor) after securing the requisite 170 credits on completion of Semester VIII							

[**Note:** Total Credit Point has been mentioned in the bracket, where 1st part indicates credits for theory and 2nd part for practical if applicable.]

Course Structure

Sem	MC / DSE	Lab	MnC	Lab	MDC	Lab	SEC	Lab	VAC
I	**MC1:** Discrete Mathematics and Graph Theory	N	**MnC1:** Programming Fundamentals using C	Y	**MDC1:** Digital Fluency (Only Practical)	Y	**SEC1:** Introduction to programming using C (Only Practical)	Y	ENVS
	MC2: Digital Logic System	Y							
II	**MC3:** Data Structures and Algorithms	Y	**MnC2:** Programming Fundamentals using C	Y	**MDC2:**		**SEC2:** Object Oriented Programming through C++	Y	Need to choose one
	MC4: Computer System Organization and Architecture	N							
III	**MC5:** Database Management System	Y	**MnC3:** Data Structures	Y	**MDC-3:**		**SEC3:** Web Technology (Only Practical)	Y	
	MC6: Operating System	Y							
IV	**MC7:** Formal Language and Automata Theory	N	**MnC4:** Data Structures	Y					
	MC8: Design & Analysis of Algorithm	Y							
	MC9: Data Communication and Networking	N							
V	**MC10:**		**MnC-5:**						Need to choose one
	MC11:								
	DSE-1:								
	DSE-2:								
VI	**MC12:**		**MnC-6:**						
	MC13:								
	DSE-3:								
	DSE-4:								
VII	**MC14:**		**MnC-7:**						Dissertation on Major OR Academic Project / Entrepreneurship
	DSE-5:								
	DSE-6:								
VIII	**MC15:**		**MnC-8:**						Dissertation on Major AND Academic Project / Entrepreneurship
	DSE-7:								
	DSE-8:								
VII	**MC14:**		**MnC-7:**						
	DSE-5:								
	DSE-6:								
	DSE-7:								
VIII	**MC15:**		**MnC-8:**						
	DSE-8:								
	DSE-9:								
	DSE-10:								

Marks Distribution

Semester	Major- MC	Lab	Minor- MnC	Lab	Multidisciplinary- MDC	Lab	SEC	Lab	VAC
I	**MC1:** TH: 40, IA:10	N	**MnC1:** TH: 25, PR: 15, IA: 10	Y	**MDC1:** PR: 40, IA:10	Y	**SEC1:** PR: 40, IA:10	Y	
	MC2: TH: 25, PR: 15, IA: 10	Y							
II	**MC3:** TH: 25, PR: 15, IA: 10	Y	**MnC2:** TH: 25, PR: 15, IA: 10	Y	**MDC2:**		**SEC2:** TH: 25, PR: 15, IA: 10	Y	
	MC4: TH: 40, IA:10	N							
III	**MC5:** TH: 25, PR: 15, IA: 10	Y	**MnC3:** TH: 25, PR: 15, IA: 10	Y	**MDC3:**		**SEC1:** PR: 40, IA:10	Y	
	MC6: TH: 25, PR: 15, IA: 10	Y							
IV	**MC7:** TH: 40, IA:10	N	**MnC4:** TH: 25, PR: 15, IA: 10	Y					
	MC8: TH: 25, PR: 15, IA: 10	Y							
	MC9: TH: 40, IA:10	N							
V	**MC10:**		**MnC5:**						
	MC11: T								
	DSE-1:								
	DSE-2:								
VI	**MC12:**		**MnC6:**						
	MC13:								
	DSE-3:								
	DSE-4:								
VII	**MC14:**		**MnC7:**						
	DSE-5:								
	DSE-6:								
VIII	**MC15:**		**MnC8:**						
	DSE-7:								
	DSE-8:								

*TH, PR, and IA stands for Theory, Practical, and Internal Assessment respectively.

Question patterns For Major Core (MC), Minor Core (MnC), Skill Enhancement Course (SEC) and Discipline Specific Elective (DSE)

- **Theory (Semester End Written Examination) -TH**

  Group-A **Group-B**

  Full Marks = 25(2 Marks x 5 Question) + (5 Marks x 3 Questions)
  Full Marks = 40(2 Marks x 5 Question) + (10 Marks x 3 Questions)

  Note: *Question(s) containing 10 marks will be divided into smaller sub-parts with maximum mark - 6. At-least one extra question will be available for each group to answer.*

- **Internal Assessment - IA**

  Full Marks = 10 Attendance (4) and Assessment (6)

- **Practical (Semester End Laboratory Based Test) - PR**

  Full Marks = 15 / 40
  [Distribution of practical examination marks have been mentioned in the concerned part of the Syllabus.]

Duration of Examination

- Theory paper of 25/40 marks: 2 hours
- Practical paper of 15 marks: 2 hours
- Practical paper of 50 marks: 4 hours

Semester-I

MC1: DMGT: Discrete Mathematics and Graph Theory **Credit -4 F.M.-50**

Sets: finite and Infinite sets, un-countably Infinite Set; Venn diagrams, set operations; functions- Domain, target, and range/image of a function - Surjections, injections, bijections, Inverses, Composition; relations- Reflexivity, symmetry, anti-symmetry, transitivity, Equivalence relations, partial orders, Properties of Binary Relations, Closure, Partial Ordering Relations;

Counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion.

Discrete Probability: Finite probability space, events, Properties of events, Conditional probability, Bayes' theorem, Independence.

Recurrences: Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Master Theorem, Growth of Functions: Asymptotic Notations.

Propositional Logic: Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.

Graph Theory : Basic Terminology, Models and Types, multi-graphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Trees, Basic Terminology and properties of Trees, Binary tree, Introduction to Spanning Tree- only basics of Kruskal and Prims algorithm.

Text/ Reference Books:

1. Elements of Discrete mathematics, C.L. Liu , D.P. Mahopatra, Tata McGraw Hill
2. Discrete Mathematics and Its Applications, Kenneth Rosen, McGraw Hill
3. Introduction to algorithms ,T.H. Coremen, C.E. Leiserson, R. L. Rivest, Prentice Hall on India,
4. Discrete Mathematics with Algorithms , M. O. Albertson and J. P. Hutchinson, John wiley Publication,
5. Discrete Structures, Logic, and Computability, J. L. Hein, , Jones and Bartlett Publishers,
6. Essentials of Discrete Mathematics, D.J. Hunter, Jones and Bartlett Publishers
7. Discrete Mathematical Structures with Applications to Combinatorics, V Ramaswamy, University Press
8. Discrete Mathematics: A Concept-based Approach, Basavaraj S Anami, Venkanna S Madalli, University Press
9. An Introduction to Probability Theory and its Applications, Vol. 1 & 2. William Feller. New York, NY: Wiley.
10. Probability Models for Computer Science. Sheldon Ross. San Diego, CA: Harcourt/Academic Press.
11. Graph Theory with Applications to Engineering & Computer Science. Narsingh Deo, Prentice Hall India
12. Introduction to Graph Theory. Douglas Brent West, Pearson Education India

MC2: DLS-a: Digital Logic System **Credit -4 F.M.-50**

Number Systems: Introduction to number system, Weighted and Non-Weighted Codes, positional, Binary, Octal, Hexadecimal, Binary coded Decimal (BCD), Gray Codes, Alphanumeric codes, ASCII, EBCDIC, Conversion of bases.

Boolean Algebra: Fundamentals of Boolean Expression: Definition of Switching Algebra, Basic properties of Switching Algebra, Huntington's Postulates, Basic logic gates (AND, OR, NOT), De-Morgan's Theorem, Universal Logic gates (NAND, NOR), Minterm, Maxterm, Minimization of Boolean Functions using K-Map, Simplification of logic expression.

Combinational Circuits: Half adders, Full Adder, Half Subtractor, Full Subtractor and construction using Basic Logic Gates (OR, AND, NOT) and Universal Logic Gates (NAND & NOR), Multibit Adder- Ripple Carry Adder, Carry Look Ahead adder, BCD Adder, Adder/Subtractor unit Construction using 4 bit Full adders units, 1 bit, 2 bit and 3 bit Comparators. Data Selector-Multiplexer: Expansion (Cascading), Function Realization. Encoders:- Realization of simple Encoders and priority Encoders using Basic and Universal Logic gates. Data Distributor:- De-multiplexer, Cascading. Chip Selector/Minterm Generator - Decoder- Function Realization, Cascading, BCD Decoders, Seven Segment Display and Decoders, realization of seven segment decoders using basic gates. Parity bit and Code Converters: Parity bit Generator/Checker, Gray to Binary code converter, Binary to Gray Code Converter.

Sequential Circuits: Set/Reset (SR) Latch: Using NAND and NOR gates, Gated S-R latches, D Latch, J-K Latch, T Flip Flop, Race around Condition, Master Slave J-K Flip Flop, Edge Triggered SR, D and JK Flip Flop, Flip-Flop Conversions, Flip-Flops with Preset and Clear.

Registers: Serial Input Serial Output, Serial Input Parallel Output, Parallel input Serial Output, Parallel Input parallel Output, Universal Shift Registers.

Counters: Asynchronous Counter: UP/DOWN Counters, Mod - N Counters, BCD Counter, Synchronous Counter: UP/DOWN Counters, Mod-N Counters, Ring Counters, Johnson Counters.

MC2: DLS-b: Digital Logic Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.

Combinational Circuits:
1. Implementation of different functions using Basic and Logic gates, SOP, POS.
2. Study and prove De-Morgan's Theorem.
3. Universal function using NAND and NOR gates
4. Implementation of half and Full adder (3-bit) using basic logic gates and Universal logic gates (NAND & NOR).
5. Implementation of half and Full Subtractor (3-bit) using basic logic gates and Universal logic gates (NAND & NOR).
6. BCD adder using 7483 and other logic gates.
7. Design 4 to 1 multiplexer using logic/Universal gates and implement full adder/full subtractor.
8. Using 74153 and 74151 to implement full adder/ full subtractor and other functions.
9. Cascading of Multiplexers.
10. Design 2 to 4 decoder using basic / universal logic gates.
11. Study 74138 and 74139 and implement full adder / full subtractor and other functions.
12. Implementation of 1 bit Comparator using decoders.
13. Cascading of Decoders.
14. Design a parity generator and checker using basic gates.
15. Construct and study comparators using 7485.
16. Construct Comparator (2-bit) using logic gates
17. Design a seven segment display unit using Common anode/Common cathode and 7447 / 7448.
18. Study Priority Encoder Chip 74147/74148.

Sequential Circuits:
1. Realization of RS, D, JK Clocked/Gated Level Triggered Flip-Flop using basic/Universal logic gates.
2. Study and Conversion of Flip-Flops: D to JK, JK to D, JK to T, SR to JK, SR to D Flip-flop.
3. Design synchronous and asynchronous counters MOD-n (MOD-8, MOD-10) UP/ DOWN and connecting Seven Segment Display along with decoder for display of counting sequence.
4. Construction of ODD/EVEN 4 bit Synchronous Counter.
5. 2-bit Universal Shift Register.

Text/Reference Books:

1. Digital Circuits, Vol - I & II, D. Ray Chaudhuri, Platinum Publishers.
2. Digital Systems - Principle & Applications, Tocci & Widmer, EEE.
3. Digital Logic & State Machine Design, Comer, Oxford.
4. Digital Principle & Applications, Malvino & Leach, McGraw Hill.
5. Digital Design, Mano, PHI.
6. Digital Integrated Electronics- H.Taub & D.Shilling, Mc Graw Hill.
7. Digital Circuits and Design, Salivahan, Vikas.
8. Digital Fundamentals, T. L. Floyd, Pearson.

Note: *Student must be familiar with Trainer-kit as well as separate breadboard, power supply, LED, Resistance, Clock based individual system. For evaluation of this paper student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Implementation – 9*
b. *Viva- Voce – 4*
c. *Lab Assignment – 2*

MnC1: PFC-a: Programming Fundamentals using C Credit -4 F.M.-50

Introduction: Basic Structure, Algorithms, Flowcharts, Structured programming constructs.
C Programming elements: Character sets, Keywords, Constants, Variables, Data Types, Operators- Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional Operator, Precedence and Associations; Expressions, type casting. Comments, Functions, Storage Classes, Bit manipulation, Input and output.
C Pre-processor: File inclusion, Macro substitution. **Statements:** Assignment, Control statements- if, if else, switch, break, continue, goto, Loops-while, do_while, for. **Functions:** Argument passing, return statement, return values and their types, recursion **Arrays:** String handling with arrays, String handling functions.1D Arrays, 2D Arrays with static and dynamic memory allocations. **Pointers:** Definition and initialization, Pointer arithmetic, Pointers and arrays,

String functions and manipulation, Dynamic storage allocation. **User defined Data types:** Structures. Structure arrays, Pointers to Functions and Structures, Unions **File Access:** Opening, Closing, I/O operations.

MnC1: PFC-b: Programming Fundamentals using C Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C compiler **(GCC)**.

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first 'n' terms of the following series, S =1+1/2+1/3+1/4+……
4. WAP to compute the sum of the first 'n' terms of the following series, S =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
*******
```

10. WAP to perform following actions on an array entered by the user : i) Print the even-valued elements. ii) Print the odd-valued elements. iii) Calculate and print the sum and average of the elements of array.
iv) Print the maximum and minimum element of array v) Remove the duplicates from the array vi) Print the array in reverse order The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. 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.
15. 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.
16. Write a menu driven program to perform following operations on strings: a) Show address of each character in string. b) Concatenate two strings without using strcat function. c) Concatenate two strings using strcat function. d) Compare two strings. e) Calculate length of the string (use pointers). f) Convert all lowercase characters to uppercase. g) Convert all uppercase characters to lowercase. h) Calculate number of vowels. i) Reverse the string.
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration.
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration.
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

Text/ Reference Books:

1. Programming with C, Byron S. Gottfried, McGraw Hill.
2. The C Programming Language, Kernighan and Dennis, PHI.
3. The Complete reference C, Herbert Schildt, McGraw Hill.
4. Let Us C, Kanitkar, BPB Publication.
5. Programming in ANSI C, Balaguruswamy, McGraw Hill.
6. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

Note: *For evaluation of this paper student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 2*
c. *Output – 5*

d. Viva- Voce – 4
e. Lab Assignment – 2

MDC1: DF: Digital Fluency (Only Practical) **Credit -3 F.M.-50**

Knowing computer: What is Computer, Basic Applications of Computer; Components of Computer System, Central Processing Unit (CPU), VDU, Keyboard and Mouse, Other input/output Devices, Computer Memory, Concepts of Hardware and Software; Concept of Computing, Data and Information; Applications of IECT; Connecting keyboard, mouse, monitor and printer to CPU and checking power supply.
Operating Computer using GUI Based Operating System: What is an Operating System; Basics of Popular Operating Systems; The User Interface, Using Mouse; Using right Button of the Mouse and Moving Icons on the screen, Use of Common Icons, Status Bar, Using Menu and Menu-selection, Running an Application, Viewing of File, Folders and Directories, Creating and Renaming of files and folders, Opening and closing of different Windows; Using help; Creating Short cuts, Basics of O.S Setup; Common utilities.
Understanding Word Processing: Word Processing Basics; Opening and Closing of documents; Text creation and Manipulation; Formatting of text; Table handling; Spell check, language setting and thesaurus; Printing of word document.
Using Spread Sheet: Basics of Spreadsheet; Manipulation of cells; Formulas and Functions; Editing of Spread Sheet, printing of Spread Sheet.
Making Small Presentation: Basics of presentation software; Creating Presentation; Preparation and Presentation of Slides; Slide Show; Taking printouts of presentation / handouts.
Introduction to Internet, WWW and Web Browsers: Basic of Computer networks; LAN, WAN; Concept of Internet; Applications of Internet; connecting to internet; What is ISP; Knowing the Internet; Basics of internet connectivity related troubleshooting, World Wide Web; Web Browsing softwares, Search Engines; Understanding URL; Domain name; IP Address; Using e-governance website.
Communications and collaboration: Basics of electronic mail; Getting an email account; Sending and receiving emails; Accessing sent emails; Using Emails; Document collaboration; Instant Messaging; etc.
Google Drive- uploading and sharing of files and folders, working with Google Meet.
E-learning -Introduction to e-learning platforms such as Swayam and MOOC. **E-Commerce**: Basic Web Commerce Concept, E- payment methods: E-cash Payment System, Credit Payment System, Types of Electronic Payment Systems: Credit Card, Debit Card; E-Money-Electronic Fund Transfer (EFT).
Introductory Overview of Emerging Technologies:
i. Artificial Intelligence, Machine Learning, Deep Learning,
ii. Database Management for Data Science, Big Data Analytics,
iii. Internet of Things (IoT) and Industrial Internet of Things (IIoT)
iv. Cloud computing and its service models
v. Cyber Security and Types of cyber attack

<u>The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course.</u>

- Identifying the configuration and version of a computer system (PC), laptop, and a mobile phone.
- Finding the background and foreground processes on Task manager.
- Translating Bengali word into English in Google embedded with AI.
- Use Google assistant on any android smartphone to dictate commands and to launch apps
- Downloading your e-aadhar.
- Creating resume in Word processor.
- Creating power-point presentation for your college introduction and apply transitions and animations.
- Create your marksheet in Microsoft Excel.
- Simple computation using spread sheet.
- Create an email-ID and sending and forwarding.
- Attaching files and downloading files in email.
- Creating a Google form for student data collection and send it to two users.
- Scheduling a virtual meet and invite peoples to join the Google meet.
- Creating a hotspot from a mobile phone, and allowing others to use the hotspot.
- Sign in and create account e-learning platforms such as Swayam and MOOC.
- Creating an account in the railway reservation website, IRCTC, and finding trains from Malda to Kolkata.
- Demo of online order placing for books using flipkart/ amazon e-commerce platform etc.
- Install any antivirus app in your system and scan.

• Demonstrate unsecured (HTTP) and secured (HTTPS) websites.
• File sharing in social media platform.

Text/ Reference Books:

1. Fundamentals of computers - V. Rajaraman - Prentice- Hall of India.
2. Computer Fundamentals - P. K. Sinha Publisher: BPB Publications.
3. Digital Fluency- Understanding the Emerging Technologies. Dr. K Venkata Nagendra, Blue ink publisher.

Note: *Evaluation of this paper will only be done through practical examination. Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Output – 25*
b. *Viva- Voce – 10*
c. *Lab Assignment – 5*

SEC1: IPC: Introduction to programming using C (Only Practical) Credit -3 F.M.-50

Introduction: Basic Structure, Algorithms, Flowcharts, Structured programming constructs.
C Programming elements: Character sets, Keywords, Constants, Variables, Data Types, Operators- Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional Operator, Precedence and Associations; Expressions, type casting. Comments, Functions, Storage Classes, Bit manipulation, Input and output.
C Pre-processor: File inclusion, Macro substitution. **Statements:** Assignment, Control statements- if, if else, switch, break, continue, goto, Loops-while, do_while, for. **Functions:** Argument passing, return statement, return values and their types, recursion **Arrays:** String handling with arrays, String handling functions.1D Arrays, 2D Arrays with static and dynamic memory allocations. **Pointers:** Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation. **User defined Data types:** Structures. Structure arrays, Pointers to Functions and Structures, Unions **File Access:** Opening, Closing, I/O operations.

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C compiler **(GCC)** for practical.

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first 'n' terms of the following series, S =1+1/2+1/3+1/4+……
4. WAP to compute the sum of the first 'n' terms of the following series, S =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9.WAP to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
```

10.WAP to perform following actions on an array entered by the user : i) Print the even-valued elements. ii) Print the odd-valued elements. iii) Calculate and print the sum and average of the elements of array.
iv) Print the maximum and minimum element of array v) Remove the duplicates from the array vi) Print the array in reverse order The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. 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.
15. 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.
16. Write a menu driven program to perform following operations on strings: a) Show address of each character in string. b) Concatenate two strings without using strcat function. c) Concatenate two strings using strcat function. d) Compare two strings. e)

Calculate length of the string (use pointers). f) Convert all lowercase characters to uppercase. g) Convert all uppercase characters to lowercase. h) Calculate number of vowels. i) Reverse the string.
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration.
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration.
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

Text/ Reference Books:

1. Programming with C, Byron S. Gottfried, McGraw Hill.
2. The C Programming Language, Kernighan and Dennis, PHI.
3. The Complete reference C, Herbert Schildt, McGraw Hill.
4. Let Us C, Kanitkar, BPB Publication.
5. Programming in ANSI C, Balaguruswamy, McGraw Hill.
6. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

Note: *Evaluation of this paper will only be done through practical examination. Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 5*
b. *Program Code – 5*
c. *Output – 15*
d. *Viva- Voce – 10*
e. *Lab Assignment – 5*

Semester – II

MC3: DSA-a: Data Structures & Algorithms **Credit -4 F.M.-50**

Introduction to Data Structure: Abstract Data Type.
Arrays: 1D, 2D and Multi-dimensional Arrays, Sparse Matrices. Polynomial representation (Polynomial Representation as Application).
Linked Lists: Singly, Doubly and Circular Lists; Polynomial representation (Polynomial Representation as Application).
Stacks: Implementing single / multiple stacks in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack.
Queues: Array and Linked representation of Queue, Circular Queue, De-queue, Priority Queues.
Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation).
Trees: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Tree), Binary Search Trees; Height-Balanced Trees (Various operations on AVL Trees), Threaded Binary Trees, B-Tree.
Searching and Sorting: Linear Search, Binary Search, Comparison of Linear and Binary Search.
Sort: Bubble sort, Selection Sort, Insertion Sort, Merge Sort, Quick sort, Heap Sort, Comparison of Sorting Techniques.
Hashing: Introduction to Hashing, Choosing a Hash Function, collision resolution techniques.

MC3: DSA-b: Data Structures & Algorithms Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C compiler **(GCC)** for practical.

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. 2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort, Selection sort etc.
3. Implement Singly Linked List. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists.
4. Implement Doubly Linked List. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation.
8. Perform Queues operations using Circular Array implementation.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. WAP to display Fibonacci series (i)using recursion, (ii) using iteration.
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion.
14. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive and Iterative Implementation) (b) Deletion by copying (c) Deletion by Merging (d) Search a no. in BST (e) Display its preorder, postorder and inorder traversals Recursively (f) Display its preorder, postorder and inorder traversals Iteratively (g) Display its level-by-level traversals (h) Count the non-leaf nodes and leaf nodes (i) Display height of tree
(j) Create a mirror image of tree (k) Check whether two BSTs are equal or not
15. WAP to reverse the order of the elements in the stack using additional stack.

Text/ Reference Books:

1) Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Pr.
2) Data Structures: A Pseudocode Approach with C, Richard F. Gilberg and Behrouz A.Forouzan, Cengage Learning
3) Data Structures In C, Noel Kalicharan, CreateSpace Independent Publishing Platform.
4) Data Structures and algorithm in C, Adam Drozdek, Cengage Learning.
5) The C Programming Language, Brian W. Kernighan and Dennis Ritchie, PrenticeHall.
6) Data Structures Using C and C++, Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, PHI.
7) Classic Data Structures, Debasis Samanta, PHI
8) Fundamental of Computer Algorithms, Horowitz, Sahni, Rajasekaran, Universities Press.

Note: *For evaluation of this paper student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 2*
c. *Output – 5*
d. *Viva- Voce – 4*
e. *Lab Assignment – 2*

MC4: CSOA: Computer System Organization and Architecture **Credit -4 F.M.-50**

Basic Structure of Computers: Basic Functional Units, Basic Operational Concept, Bus Structure, Software, Performance, Multiprocessor and Multicomputer.
Register Transfer and Micro-operation: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Three State Bus Buffers, memory Transfer, Arithmetic and Logical micro-operations, Shift and Arithmetic shifts.
Basic Computer Organization and Design: Instruction Codes, Stored Program Organization, Indirect Address, Computer Registers, Common Bus System, Computer Instruction, Timing and Control, Instruction Cycle, fetch Decode, Register Reference Instructions, Memory Reference Instruction, Input-Output and Interrupt, Design of Basic Computer, Design of Accumulator Logic.
CPU Organization: Arithmetic and Logic Unit (ALU)- Combinational ALU, 2'S Complement Addition, Subtraction Unit, Booths Algorithm for Multiplication, Division Hardware using Restoration Division Algorithm. General register organization, Accumulator Based, Register Based, Stack Type CPU organization.

Control Unit: Hardwired Control Unit, Micro-programmed Control Unit: Control memory, Address Sequencing, conditional branching, mapping of instructions, subroutine, Design of Control Unit.
CPU Registers: Program Counter, Stack Pointer Register, Memory Address Register, Instruction Register, Memory Buffer Register, Flag registers, Temporary Registers.
Instructions: Operational Code, Operands, Zero, One, Two and Three Address Instruction, Instruction Types, Addressing modes, Data Transfer and Manipulation instructions, Program control instructions.
CISC and RISC processors: Introduction, relative merits and De-merits.
Input / Output Organization: Polling, Interrupts, subroutines, Memory mapped IO, IO mapped IO, DMA, Bus Arbitration.
Memory: Primary memory: ROM, PROM, EPROM, EEPROM, Flash memory, RAM: SRAM,DRAM, Cache Memory: Mapping Functions, Replacement Algorithms, Hit and Miss ratio, Virtual memories, Address Translation, Memory Management requirements, Secondary Storage: Magnetic Hard Disks.

Text/Reference Books:

1. Computer System Architecture, Morries Mano, Pearson.
2. Computer Organization & Architecture, Williams Stallings, Pearson.
3. Computer Organization, Hamacher, Vranesic and Zaky, McGraw Hill.
4. Computer Architecture and Organization, Govindrajalu, Tata McGraw Hill.
5. Computer Architecture and Organization, J P Hayes, Tata McGRaw Hill.
6. Structured Computer Organization, Andrew S. Tanenbaum, Austin, Pearson.

MnC2: PFC-a: Programming Fundamentals using C **Credit -4 F.M.-50**

Introduction: Basic Structure, Algorithms, Flowcharts, Structured programming constructs.
C Programming elements: Character sets, Keywords, Constants, Variables, Data Types, Operators- Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional Operator, Precedence and Associations; Expressions, type casting. Comments, Functions, Storage Classes, Bit manipulation, Input and output.
C Pre-processor: File inclusion, Macro substitution. **Statements:** Assignment, Control statements- if, if else, switch, break, continue, goto, Loops-while, do_while, for. **Functions:** Argument passing, return statement, return values and their types, recursion **Arrays:** String handling with arrays, String handling functions.1D Arrays, 2D Arrays with static and dynamic memory allocations. **Pointers:** Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation. **User defined Data types:** Structures. Structure arrays, Pointers to Functions and Structures, Unions **File Access:** Opening, Closing, I/O operations.

MnC2: PFC-b: Programming Fundamentals using C Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C compiler **(GCC).**

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first 'n' terms of the following series, S =1+1/2+1/3+1/4+……
4. WAP to compute the sum of the first 'n' terms of the following series, S =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9.WAP to print a triangle of stars as follows (take number of lines from user):

*
***
*****
****

10.WAP to perform following actions on an array entered by the user : i) Print the even-valued elements. ii) Print the odd-valued elements. iii) Calculate and print the sum and average of the elements of array.
iv) Print the maximum and minimum element of array v) Remove the duplicates from the array vi) Print the array in reverse order The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.

14. 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.
15. 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.
16. Write a menu driven program to perform following operations on strings: a) Show address of each character in string. b) Concatenate two strings without using strcat function. c) Concatenate two strings using strcat function. d) Compare two strings. e) Calculate length of the string (use pointers). f) Convert all lowercase characters to uppercase. g) Convert all uppercase characters to lowercase. h) Calculate number of vowels. i) Reverse the string.
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration.
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration.
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

Text/ Reference Books:

1. Programming with C, Byron S. Gottfried, McGraw Hill.
2. The C Programming Language, Kernighan and Dennis, PHI.
3. The Complete reference C, Herbert Schildt, McGraw Hill.
4. Let Us C, Kanitkar, BPB Publication.
5. Programming in ANSI C, Balaguruswamy, McGraw Hill.
6. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

Note: *For evaluation of this paper student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 2*
c. *Output – 5*
d. *Viva- Voce – 4*
e. *Lab Assignment – 2*

SEC2: OOP-a: Object Oriented Programming through C++ Credit -3 F.M.-50

Introduction to C++: Overview of Procedural Programming and Object-Orientation Programming, Using main() function, Compiling and Executing Simple Programs in C++. Concepts of Data Types, Variables, Constants, Operators and Basic I/O Expressions, Conditional Statements and Iterative Statements, Functions and Arrays Pointers and References in C++, Memory Allocation in C++.
Using Classes in C++: Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables &Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors.
Overview of Function Overloading and Operator Overloading: Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (unary and binary operators)
Inheritance, Polymorphism, Exception Handling and Generic function: Introduction to Inheritance (Single, Multi-Level, Multiple, Hierarchical, Hybrid), Polymorphism (Virtual Functions, Pure Virtual Functions), Abstract class, Basics of Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions. Generic function- Template class and function.

SEC2: OOP-b: Object Oriented Programming through C++ Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C++ compiler **(GNU C++)**.

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first 'n' terms of the following series S = 1+1/2+1/3+1/4+……

4. WAP to compute the sum of the first ‘n’ terms of the following series S =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):
*
***
*****
****
10. WAP to perform following actions on an array entered by the user: i.Print the even-valued elements ii.Print the odd-valued elements iii.Calculate and print the sum and average of the elements of array iv.Print the maximum and minimum element of array v.Remove the duplicates from the array vi.Print the array in reverse order The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. 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.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using new operator.
16. Write a menu driven program to perform following operations on strings: a) Show address of each character in string b) Concatenate two strings without using strcat function. c) Concatenate two strings using strcat function. d) Compare two strings e) Calculate length of the string (use pointers) f) Convert all lowercase characters to uppercase g) Convert all uppercase characters to lowercase
h) Calculate number of vowels i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array. 18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
24. Create a class Box containing length, breath and height. Include following methods in it: a) Calculate surface Area b) Calculate Volume c) Increment, Overload ++ operator (both prefix & postfix) d) Decrement, Overload -- operator (both prefix & postfix) e) Overload operator == (to check equality of two boxes), as a friend function f) Overload Assignment operator g) Check if it is a Cube or cuboid Write a program which takes input from the user for length, breath and height to test the above class.
25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
26. Write a program to retrieve the student information from file created in previous question and print it in following format: Roll No. Name Marks
27. Copy the contents of one text file to another file, after removing all whitespaces.
28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

Text/ Reference Books:

1. C++: The Complete Reference, Herbtz Schildt, McGraw Hill.
2. The C++ Programming Language, Bjarne Stroustrup, Addison-Wesley.
3. Programming -- Principles and Practice using C++,Bjarne Stroustroup, Addison-Wesley.
4. Object Oriented Programming with C++, E Balaguruswamy, Tata McGraw-Hill Education.
5. C++ How to Program, Paul Deitel, Harvey Deitel, Prentice Hall.
6. Programming with C++, John R. Hubbard, Schaum's Series.
7. Accelerated C++, Andrew Koeni, Barbara, E. Moo, Published by Addison-Wesley.
8. Effective C++, Scott Meyers, Published by Addison-Wesley.
9. Head First C++ Programming: The Definitive Beginner's Guide, Harry, H. Chaudhary, First Create space Inc, O-D Publishing, LLC USA.
10. Problem Solving with C++, Walter Savitch, Pearson Education.
11. C++ Primer, Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, Published by Addison-Wesley.

Note: *Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 2*
c. *Output – 5*
d. *Viva- Voce – 4*
e. *Lab Assignment – 2*

Semester – III

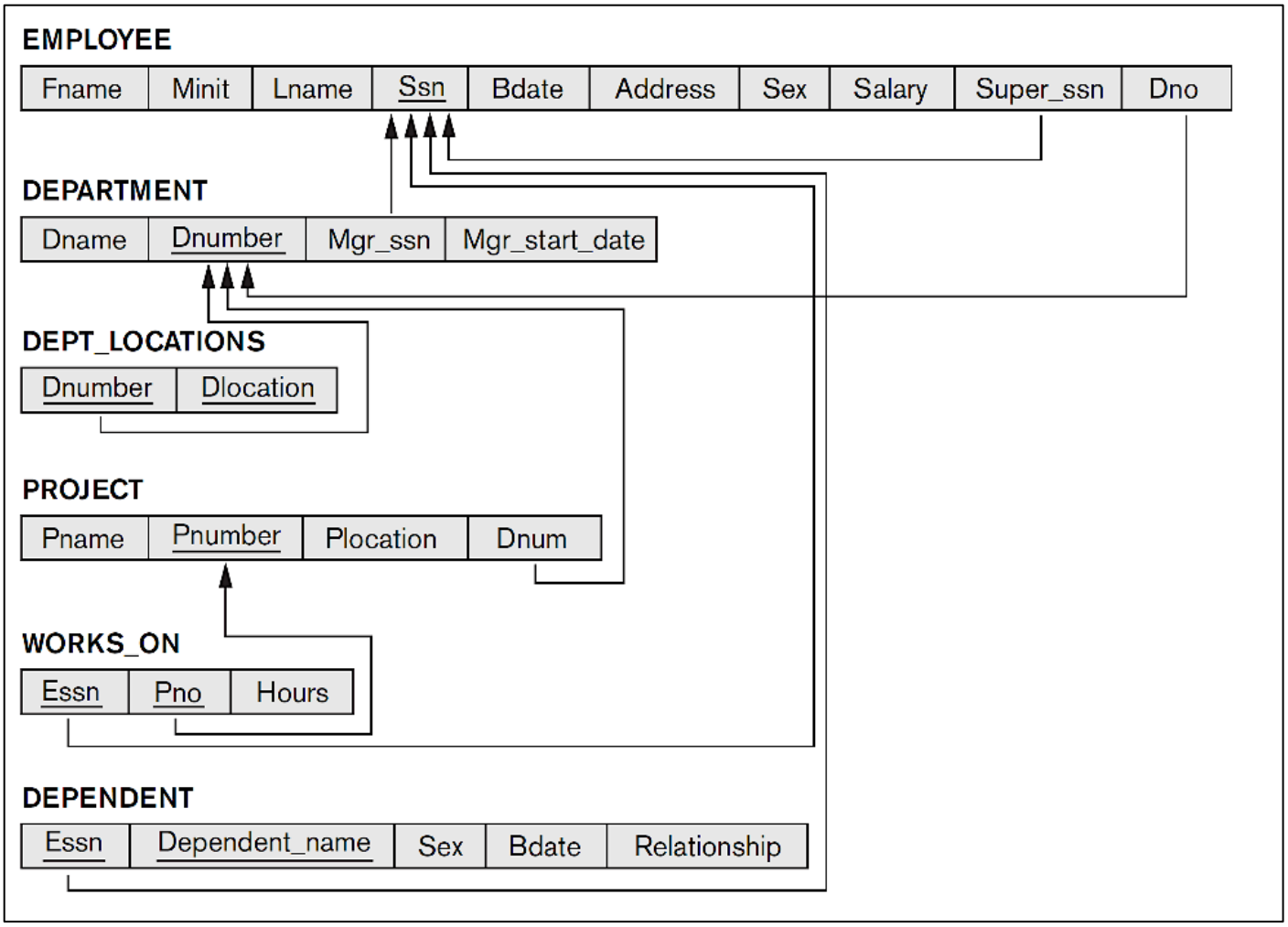
MC5: DBMS-a: Database Management System — Credit -4 F.M.-50

Introduction: Drawbacks of file System; Advantages of DBMS; Layered Architecture of Database, Data Independence; Data Models; Schemas And Instances; Database Languages; Database Users, DBA; Data Dictionary; Functional Components of a DBMS.
Entity Relationship(ER) Modelling: Entity, Attributes and Relationship, Structural Constraints, Keys, ER Diagram of Some Example Database, Weak Entity Set, Specialization and Generalization, Constraints of Specialization and Generalization, Aggregation.
Relational Model: Basic Concepts of Relational Model; Relational Algebra.
Integrity Constraints: Domain Constraints, Referential Integrity, Assertions, Triggers.
Relational Database Design: Problems of Un-Normalized Database; Functional Dependencies (FD), Derivation Rules, Closure of FD Set, Membership of A Dependency, Canonical Cover; Decomposition to 1NF, 2NF, 3NF and BCNF Using FD; Lossless Join Decomposition Algorithm; Dependency preservation. **SQL:** Basic Structure, Data Definition, Constraints and Schema Changes; Basic SQL Queries (Selection, Insertion, Deletion, Update); Order by Clause; Complex Queries, Aggregate Function and Group by Clause; Nested Sub Queries; Correlated Sub Queries; Views (Insert-Able and Updatable), Joined Relations; Set Comparisons (All, Some).
Record Storage and File Organization (Concepts only): Fixed Length and Variable Length Records; Spanned and Un-Spanned Organization of Records; Primary File Organizations and Access Structures Concepts; Unordered, Sequential, Hashed; Concepts of Primary and Secondary Index; Dense and Sparse Index; Index Sequential Files; Multilevel Indices.
Transaction Processing (Concepts only): ACID Properties; Transaction States, Concurrent Execution; Serializability (Conflict and View), Recoverability, Test for Serializability.

MC5: DBMS-b: Database Management System Practical

Some sample examples are given below. More problems can be included related to the theory. Use open source MYSQL for practical.

DDL Commands • Create table, alter table, drop table.
DML Commands • Select , update, delete, insert statements • Condition specification using Boolean and comparison operators (and, or, not,=,<>,>,=,<=) • Arithmetic operators and aggregate functions(COUNT, SUM, AVG, MIN, MAX) • Multiple table queries (join on different and same tables) • Nested select statements • Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists, union, intersect, minus, etc.) • Categorization using group by, having • Arranging using order by
Relational Database Schema – Company



Questions to be performed on above schema

A. Create tables with relevant foreign key constraints and other constraints. Populate the tables with data. Perform the following queries on the database :

1. Display all the details of all employees working in the company. **2.** Display ssn, lname, fname, address of employees who work in department no 7. **3.** Retrieve the birthdate and address of the employee whose name is 'Franklin T. Wong'. **4.** Retrieve the name and salary of every employee. **5.** Retrieve all distinct salary values. **6.** Retrieve all employee names whose address is in 'Bellaire'. **7.** Retrieve all employees who were born during the 1950s. **8.** Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive). **9.** Retrieve the names of all employees who do not have supervisors. **10.** Retrieve SSN and department name for all employees. **11.** Retrieve the name and address of all employees who work for the 'Research' department. **12.** For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date. **13.** For each employee, retrieve the employee's name, and the name of his or her immediate supervisor. **14.** Retrieve all combinations of Employee Name and Department Name. **15.** Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project. **16.** Increase the salary of all employees working on the 'ProductX' project by 15%. Retrieve employee name and increased salary of these employees. **17.** Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name. **18.** Select the names of employees whose salary does not match with salary of any employee in department 10. **19.** Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee. **20.** Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford. **21.** Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings. **22.** Find the sum of the salaries and number of employees of all employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department. **23.** Select the names of employees whose salary is greater than the average salary of all employees in department 10. **24.** For each department, retrieve the department number, the number of employees in the department, and their average salary. **25.** For each project, retrieve the project number, the project name, and the number of employees who work on that project. **26.** Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively. **27.** For each department having more than 10 employees, retrieve the department no, no of employees drawing more than 40,000 as salary. **28.** Insert a record in Project table which violates referential integrity constraint with respect to Department number. Now remove the violation by making necessary insertion in the Department table. **29.** Delete all dependents of employee whose ssn is '123456789'. **30.** Delete an employee from Employee table with ssn = '12345'(make sure that this employee has some dependents, is working on some project, is a manager of some department and is supervising some employees). Check and display the cascading effect on Dependent and Works on table. In Department table MGRSSN

should be set to default value and in Employee table SUPERSSN should be set to NULL. **31.** Perform a query using alter command to drop/add field and a constraint in Employee table.

Text/ Reference Books:

1. Fundamentals of Database Systems 6th Edition, R. Elmasri, S.B. Navathe, Pearson Education.
2. Database Management Systems, R. Ramakrishanan, J. Gehrke,3rd Edition, McGraw-Hill.
3. Database System Concepts 6th Edition, A. Silberschatz, H.F. Korth, S. Sudarshan, McGraw Hill.
4. Database Systems Models, Languages, Design and application Programming, R. Elmasri, S.B. Navathe, Pearson Education.
5. SQL and Relational Theory: How to Write Accurate SQL Code, Christopher J. Date, O'Reilly Media.
6. Database Systems: A Practical Approach to Design, Implementation and Management, Thomas M. Connolly and Carolyn E. Begg, Pearson.

Note: *Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Program/Query – 9*
b. *Viva- Voce – 4*
c. *Lab Assignment – 2*

MC6: OS-a: Operating System **Credit -4 F.M.-50**

Introduction: Basic OS functions, types of operating systems: batch systems–multiprogramming systems, time sharing systems;
Operating System Organization: Processor and user modes, kernels, system calls and system programs.
Process: System view of the process and resources, process hierarchy, threads, threading issues.
Process Scheduling: Scheduling criteria, Pre-emptive and non-preemptive scheduling, Long term, short term and medium term, FCFS, SJF, SRTF, Priority scheduling, Round Robin, Multilevel Queue Scheduling, Multilevel Queue Feedback Scheduling.
Process Synchronization: Concurrent Processes, critical section, semaphores and application, methods for inter-process communication;
Deadlock: Definition, Prevention, Avoidance, Detection, Recovery, Banker's algorithm.
Memory Management: Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory
File and I/O Management: Directory structure, file operations, file allocation methods, disc management.

MC6: OS-b: Operating System Practical

Some sample examples/Commands are given below. More problems can be included related to the theory. Use open source system (Debian OS) for practical.

1. Usage of following commands: ls, pwd, tty, cat, who, who am i, rm, mkdir, rmdir, touch, cd.
2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, tput (clear, highlight), bc.
4. Write a shell script to check if the number entered at the command line is prime or not.
5. Write a shell script to modify "cal" command to display calendars of the specified months.
6. Write a shell script to modify "cal" command to display calendars of the specified range of months.
7. Write a shell script to accept a login name. If not a valid login name display message – "Entered login name is invalid".
8. Write a shell script to display date in the mm/dd/yy format.
9. Write a shell script to display on the screen sorted output of "who" command along with the total number of users .
10. Write a shell script to display the multiplication table any number,
11. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
12. Write a shell script to find the sum of digits of a given number.
13. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
14. Write a shell script to find the LCD(least common divisor) of two numbers.
15. Write a shell script to perform the tasks of basic calculator.
16. Write a shell script to find the power of a given number.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.
20. Program to show the pyramid of special character "*".

Text/ Reference Books:

1. Operating Systems Concepts, A Silberschatz, P.B. Galvin, G. Gagne, John Wiley Publications.
2. Modern Operating Systems, A.S. Tanenbaum, Pearson Education.
3. Operating Systems: A Modern Perspective, G. Nutt, Pearson Education.
4. Operating Systems, Internals & Design Principles W.Stallings, PHI.
5. Operating Systems- Concepts and design, M. Milenkovic, Tata McGraw Hill.
6. UNIX Concepts and Applications, Sumitabha Das , Tata McGraw-Hill.
7. Understanding the Linux Kernel, D. P. Bovet and M. Cesati, O'Reilly.

Note: *Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Program – 9*
b. *Viva- Voce – 4*
c. *Lab Assignment – 2*

MnC3: DS-a: Data Structures **Credit -4 F.M.-50**

Introduction to Data Structure: Abstract Data Type.
Arrays: 1D, 2D arrays, Sparse Matrices.
Linked Lists: Singly, basics of Doubly and Circular Lists; Polynomial representation (Polynomial Representation as Application).
Stacks: Implementing stacks using Array and Linked list; Prefix, Infix and Postfix expressions, Infix to postfix conversion; Applications of stack.
Queues: Array and Linked representation of Queue, basics of Circular Queue.
Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation).
Trees: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Trees), Binary Search Trees; Basic knowledge of Height-Balanced Trees.
Searching and Sorting: Linear Search, Binary Search, Comparison of Linear and Binary Search.
Sort: Bubble sort, Selection Sort, Insertion Sort, Merge Sort, and Quick sort, Comparison of Sorting Techniques.
Hashing: Introduction to Hashing, Choosing a Hash Function, collision resolution techniques.

MnC3: DS-b: Data Structures Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C compiler **(GCC)** for practical.

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. 2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort, Selection sort etc.
3. Implement Singly Linked List. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists.
4. Perform Stack operations using Linked List implementation.
5. Perform Stack operations using Array implementation.
6. WAP to reverse the order of the elements in the stack using additional stack.
7. Create and perform different operations on Double-ended Queues using Linked List implementation.
8. WAP to display Fibonacci series (i)using recursion, (ii) using iteration.
9. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
10.WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion.
11. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive Implementation) (b) Deletion by copying (c) Deletion by Merging (d) Search a no. in BST (e) Display its preorder, postorder and inorder traversals Recursively (f) Display its preorder, postorder and inorder traversals Iteratively (g) Display its level-by-level traversals.

Text/ Reference Books:

1) Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Pr.
2) Data Structures: A Pseudocode Approach with C, Richard F. Gilberg and Behrouz A.Forouzan, Cengage Learning
3) Data Structures In C, Noel Kalicharan, CreateSpace Independent Publishing Platform.
4) Data Structures and algorithm in C, Adam Drozdek, Cengage Learning.
5) The C Programming Language, Brian W. Kernighan and Dennis Ritchie, PrenticeHall.
6) Data Structures Using C and C++, Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, PHI.
7) Classic Data Structures, Debasis Samanta, PHI
8) Fundamental of Computer Algorithms, Horowitz, Sahni, Rajasekaran, Universities Press.

Note: *For evaluation of this paper student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 2*
c. *Output – 5*
d. *Viva- Voce – 4*
e. *Lab Assignment – 2*

SEC3: WT: Web Technology (Only Practical) **Credit -3 F.M.-50**

Introduction to HTML : The development process, Html tags and simple HTML forms, web site structure
Introduction to XHTML : XML, Move to XHTML, Meta tags, Character entities, frames and frame sets, inside browser.
Style sheets : Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS.
Javascript : Client side scripting, What is Javascript?, How to develop Javascript?, simple Javascript, variables, functions, conditions, loops and repetition. Advance script, Javascript and objects, Javascript own objects, the DOM and web browser environments, forms and validations DHTML : Combining HTML, CSS and Javascript, events and buttons, controlling your browser, Ajax: Introduction, advantages & disadvantages, Purpose of it ,ajax based web application, alternatives of ajax
XML : Introduction to XML, uses of XML, simple XML, XML key components, DTD and Schemas, Well formed, using XML with application.
PHP : Starting to script on server side, Introduction to PHP, Basic Programming Concepts of PHP: Variables, Data-types, Constants, Scope of Variables, Type of Variables, Type Casting, Operators, Operators Precedence, References, Arrays; Control Structures: Branching, If statement, Switch statement; Looping: for Loop, while Loop, do while Loop, for each Loop; Functions: User Defined Functions, Built-in Function, Arrays, function and forms, advance PHP Databases : Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHP myadmin and database bugs.

Text/ Reference Books:

1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill.
3. Web Programming, building internet applications, Chris Bates 2nd edition, Wiley Dremtech
4. Java Server Pages – Hans Bergsten, SPD O'Reilly
5. Java Script, D. Flanagan, O'Reilly, SPD.
6. Beginning Web Programming-Jon Duckett WROX.
7. Programming world wide web, R.W. Sebesta. Fourth Edition, Pearson.
8. Internet and World Wide Web – How to program, Dietel and Nieto, Pearson.

Note: *Evaluation of this paper will only be done through practical examination. Student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Output – 25*
b. *Viva- Voce – 10*
c. *Lab Assignment – 5*

Semester – IV

MC7: FLAT: Formal Language and Automata Theory **Credit -4 F.M.-50**

Finite Automata: Definition of a Finite Automaton, Model, Representation, Classification – with respect to output function Mealy and Moore Machines, with respect to State Transition – Deterministic and Non-Deterministic Machine, Examples, conversion algorithms Mealy to Moore and Moore to Mealy, Non-Deterministic to equivalent

Deterministic Finite automata, Finite and Infinite state machines, Removal of Null-transitions, Acceptability of String by a Finite Automaton, Design of different Finite State Machines, Minimized Equivalent Machine.

Formal Languages and Grammar: Introduction to Formal Grammar and Language, Formal Definition, Chomsky's Classification of Grammar – Type 0, Type-1 or Context Sensitive, Type-2 or Context Free and Type-3 or Regular Grammar, Illustration of each of these classes with example, Sentential form, Sentences – Languages or strings, Derivations – left, right derivation, Derivation tree, Parse Tree, Syntax Tree, Ambiguous Grammar and Language, Designing of Grammar for a language, Finding Language for Given Grammar; Definition and basic idea about Push Down Automaton

Regular Expression: Basic Idea and Definition, Regular Expression basic Identities, Arden's Theorem and application for reduction of equivalent regular expressions, Thompson's Construction Algorithm – Regular expression to Finite Automata conversion, State Transition System to Regular Expression conversion algorithm by Arden's Algebraic Method, FA to Regular Grammar and Regular Grammar to FA conversion algorithms and applications.

Turing Machine: Concepts of Turing Machine, Formal Definitions, Classifications – Deterministic and Non-Deterministic Turing Machines, Simple Design of Turing Machines like – Unary Adder, Subtractor, Concatenator, Odd / even count etc and concepts of Universal Turing Machines.

Text/ Reference Books:

1. Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson.
2. Theory of Computer Science (Automata, Languages & Computation) by K L P Misra & N Chandrasekharan, PHI.
3. Introduction to Theory of Computation,[Micheal Sipser, Cengage Learning.
4. Switching and Finite Automata Theory by Zvi Kohavi, Niraj. K. Jha, TMH.
5. Formal Language and Automata, P. Linz, Narosa
6. Introduction to Formal Languages, Automata Theory and Computation, Kamala Krithivasan, Pearson.
7. Elements of the Theory of Computation, Christos Papadimitriou and Harry R. Lewis, Pearson.

MC8: DAA-a: Design & Analysis of Algorithm **Credit -4 F.M.-50**

Introduction- Fundamentals of Algorithms and Analysis of Algorithms - Orders of Magnitude (Asymptotic notations); Growth rates, some common bounds (constant, logarithmic, linear, polynomial, exponential); Average and worst case analysis; Analysing control statements; Recurrence Relations- substitution, change of variables, master's method

Divide and conquer algorithms - Introduction - Quick sort, worst and average case complexity, Merge sort, Matrix multiplication, Binary search, Binary search tree

Greedy algorithms - General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm- Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem; Huffman coding

Dynamic programming – Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming- Fibonacci numbers, Warshall and Floyd algorithm; Matrix chain multiplication, Longest Common Subsequence(LCS).

Graph Algorithms: - An introduction using graphs and games - Traversing Trees: Depth First Search, Breath First Search.

Backtracking and Branch and Bound:– 0/1 Knapsack Problem, The Eight queens problem, Travelling Salesman problem.

String matching – Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, KMP Algorithm

Introduction to Complexity Theory - The class P and NP, Polynomial reduction, NP- Complete Problems, NP-Hard Problems.

MC8: DAA-b: Design & Analysis of Algorithm Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C / C++ compiler for practical **(GCC / GNU C++)**.

1. Implementation and Time analysis of sorting algorithms. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort .
2. Implementation and Time analysis of linear and binary search algorithm.
3. Write a program for Strassen's Matrix Multiplication.
4. To implement Huffman coding and analyse its time complexity
5. Write a program for travelling salesman problem.
6. Implementation of chain matrix multiplication using dynamic programming.

7. Implementation of making a change problem using dynamic programming
8. Implementation of a knapsack problem using greedy algorithm
9. Implementation of Graph and Searching (DFS and BFS).
10. Implement prim's algorithm.
11. Implement Kruskal's algorithm.
12. Implement LCS problem.
13. To implement following string matching algorithms and analyse time complexities:
    a. Naïve, b. Rabin Karp, c. Knuth Morris Pratt
14. Write a program for Floyd-Warshall algorithm.

Text/ Reference Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivestand Clifford Stein, PHI.
2. Fundamental of Algorithms by Gills Brassard, Paul Bratley, PHI.
3. Design and Analysis of Computer Algorithms by Aho, Hopcroft and Ullman, Pearson.
4. The Algorithm Design Manual By Steve s. Skiena.
5. Algorithm Design: Foundations, Analysis, and Internet Examples, Michael T. Goodrich, Wiley.
6. Design And Analysis Of Algorithms, S. Sridhar, Oxford University Press.
7. Computer Algorithms: Introduction to Design and Analysis, Sara Baase, Pearson.
8. Fundamentals of Computer Algorithms, Ellis Horowitz and Sartaj Sahni, Universities Press.
9. Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.
10. Algorithms. Sanjoy Dasgupta, Christos Papadimitriou, Vijay Vazirani, McGraw Hill.

Note: *For evaluation of this paper student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 2*
c. *Output – 5*
d. *Viva- Voce – 4*
e. *Lab Assignment – 2*

MC9: DCN: Data Communication and Networking **Credit -4 F.M.-50**

Data Communication Concepts: Analog and Digital Signals, Periodic and Non-periodic signals, Time and Frequency Domain, Bandwidth and Data rate, Signal rate, Serial and Parallel Transmission. Protocol. OSI &TCP/IP Model.
Various modes of transmission: Simplex/ Half Duplex, Duplex; Features of guided and unguided transmission media; Circuit Switching, Packet Switching; transmission impairment; Shannon Capacity.
Physical structure of Network: Types of connections (Topologies), Categories of Computer Network: LAN, MAN, WAN; Digital to Digital conversion: line coding schemes; Analog to Digital Conversion: PCM, DM; Digital to Analog conversion: ASK, PSK, FSK, QAM; Modulation and Encoding: AM, FM, PM;
Multiplexing: FDM, TDM, WDM.
Data link layer: Introduction, Framing, and Error – Detection and Correction – Parity – LRC – CRC Checksum, Hamming code, Flow and Error Control, Noiseless Channels, Noisy Channels.
Network layer: Logical Addressing, Internetworking, IP-related problems.
Transport Layer: Process to Process Delivery, UDP and TCP protocols, Data Traffic, Congestion.
Application Layer: Domain name space, DNS in internet, electronic mail, SMTP, FTP, WWW, HTTP.
Protocols: IP, ARP, RARP, TCP, UDP, SMTP, FTP, DNS, DHCP etc.
Introduction to networks and devices: Network classes, Repeaters, Hub, Bridges , Switches, Routers, Gateways; Routing Algorithms- Distance Vector Routing , Link State Routing.

Text/Reference books:

1. Data Communications and Networking, B. A. Forouzan, TMH.
2. Computer Networks, A. S. Tanenbaum, PHI.
3. Data and Computer Communication, W. Stallings, Pearson.
4. Data Communication and Networks, Bhushan Trivedi, OUP.
5. Data Communications and Computer Networks, P. C. Gupta, PHI.
6. Computer Networking: A Top-Down Approach, Kurose, J. and K. Ross, Pearson.
7. Data Communication and Computer Networks, A. Pal, PHI.
8. Understanding Data Communications, G. Held, Addison-Wesley
9. Internetworking with TCP/IP Volume I, D. E. Comer, Pearson.

MnC4: DS-a: Data Structures **Credit -4 F.M.-50**

Introduction to Data Structure: Abstract Data Type.
Arrays: 1D, 2D arrays, Sparse Matrices.
Linked Lists: Singly, basics of Doubly and Circular Lists; Polynomial representation (Polynomial Representation as Application).
Stacks: Implementing stacks using Array and Linked list; Prefix, Infix and Postfix expressions, Infix to postfix conversion; Applications of stack.
Queues: Array and Linked representation of Queue, basics of Circular Queue.
Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation).
Trees: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Trees), Binary Search Trees; Basic knowledge of Height-Balanced Trees.
Searching and Sorting: Linear Search, Binary Search, Comparison of Linear and Binary Search.
Sort: Bubble sort, Selection Sort, Insertion Sort, Merge Sort, and Quick sort, Comparison of Sorting Techniques.
Hashing: Introduction to Hashing, Choosing a Hash Function, collision resolution techniques.

MnC4: DS-b: Data Structures Practical

The assignments listed below are illustrative examples and not an exhaustive list. They serve as a starting point to cover various aspects of the course. Use open source C compiler **(GCC)** for practical.

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. 2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort, Selection sort etc.
3. Implement Singly Linked List. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists.
4. Perform Stack operations using Linked List implementation.
5. Perform Stack operations using Array implementation.
6. WAP to reverse the order of the elements in the stack using additional stack.
7. Create and perform different operations on Double-ended Queues using Linked List implementation.
8. WAP to display Fibonacci series (i)using recursion, (ii) using iteration.
9. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
10.WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion.
11. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive Implementation) (b) Deletion by copying (c) Deletion by Merging (d) Search a no. in BST (e) Display its pre-order, post-order and in-order traversals Recursively (f) Display its pre-order, post-order and in-order traversals Iteratively (g) Display its level-by-level traversals.

Text/ Reference Books:

1) Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Pr.
2) Data Structures: A Pseudocode Approach with C, Richard F. Gilberg and Behrouz A.Forouzan, Cengage Learning
3) Data Structures In C, Noel Kalicharan, CreateSpace Independent Publishing Platform.
4) Data Structures and algorithm in C, Adam Drozdek, Cengage Learning.
5) The C Programming Language, Brian W. Kernighan and Dennis Ritchie, PrenticeHall.
6) Data Structures Using C and C++, Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, PHI.
7) Classic Data Structures, Debasis Samanta, PHI
8) Fundamental of Computer Algorithms, Horowitz, Sahni, Rajasekaran, Universities Press.

Note: *For evaluation of this paper student has to submit a lab notebook properly signed by the concerned faculty member of the institution with departmental stamp. Distribution of marks will be as per the following-*

a. *Algorithm / Flow Chart – 2*
b. *Program Code – 2*
c. *Output – 5*
d. *Viva- Voce – 4*
e. *Lab Assignment – 2*