Proposed Syllabus

For

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

Submitted

To

University of Gour Banga

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

Computer Science Course Structure

Semester	Major- MC	Lab	Minor- MnC	Lab	Multidisciplinary-MDC	Lab	SEC	Lab
I	**MC1:** Discrete Mathematics(4)	N	**MnC-1:** Programming Fundamentals using C (3+1=4)	Y	**MDC-1:** Fundamentals of Computer Application (Only Practical) (3)	Y	**SEC-1:** Introduction to programming using C (Only Practical) (3)	Y
	MC2: Digital Logic System (3+1=4)	Y						
II	**MC3:** Data Structures & Algorithms (3+1=4)	Y	**MnC-2:** Programming Fundamentals using C (3+1=4)	Y	**MDC-2:** Computational Statistics (3)	N	**SEC-2:** Object Oriented Programming through C++ (2+1=3)	Y
	MC4: Computer System Architecture (4)	N						

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

Computer Science Marks Distribution

Semester	Major- MC	Lab	Minor-MnC	Lab	Multidisciplinary-MDC	Lab	SEC	Lab
I	**MC1:** TH: 40, IA:10	N	**MnC-1:** TH: 25, PR: 15, IA: 10	Y	**MDC-1:** PR: 50	Y	**SEC-1:** PR: 50	Y
	MC2: TH: 25, PR: 15, IA: 10	Y						
II	**MC3:** TH: 25, PR: 15, IA: 10	Y	**MnC-2:** TH: 25, PR: 15, IA: 10	Y	**MDC-2:**TH: 40, IA: 10	N	**SEC-2:** TH: 25, PR: 15, IA: 10	Y
	MC4: TH: 40, IA:10	N						

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

Descriptive Type Question patterns For Major Core (MC), Skill Enhancement Course (SEC) and Discipline Specific Core (DSC)

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

	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 must be set for each group.

- **Internal Assessment**

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

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

  Full Marks = 15 / 50
  Mentioned in the concerned part of the syllabus

Duration of Examination

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

Semester-I

MC1-DM: Discrete Mathematics: Credit -4 F.M.-50

Sets: finite and Infinite sets, un-countably Infinite Set; functions, relations, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion.

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

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- Kruskal and Prims algorithm.

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

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

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 Lab:

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. 1 Digit 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

Note: Student must be familiar with Trainer-kit as well as separate breadboard, power supply, LED, Resistance, Clock based system.

MnC-1-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.

MnC-1-PFC-b: Programming Lab (using C)

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 submitted a lab notebook properly signed by the concerned faculty member 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

MDC-1-FCA: Fundamentals of Computer Applications (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.

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

a. Output – 20
b. Viva- Voce – 20
c. Lab Assignment – 10

SEC-1-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.

Note: Evaluation of this paper will only be done through practical examination. Student has to submitted a lab notebook properly signed by the concerned faculty member 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 – 20
e. Lab Assignment – 5

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

Semester – II

MC-3-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 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.

MC-3-DSA-b: Data Structures & Algorithms

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 using templates. 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, Algorithms and applications in C++, Sartaj Sahni,Universities Press.
7) Data Structures Using C and C++, Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, PHI. 8) Classic Data Structures, Debasis Samanta, PHI
9) Fundamental of Computer Algorithms, Horowitz, Sahni, Rajasekaran, Universities Press.

Note: For evaluation of this paper student has to submitted a lab notebook properly signed by the concerned faculty member 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-CSA: Computer System 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.

MnC-2-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.

MnC-2-PFC-b: Programming Lab (using C)

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 submitted a lab notebook properly signed by the concerned faculty member 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

MDC-2-CST: Computational Statistics Credit -3 F.M.-50

Introduction and overview, Concepts and significance of statistics in Computer Science,
Data and Types of Measurement.

Primary and Secondary Data: Merits and Demerits; Methods of Collecting Primary Data,
Population and Sample: Merits and Demerits, Sampling Methods: Random and Non-Random,
Sampling Size and Distribution, Sampling and Non - Sampling Errors.
Methods of Collecting Secondary Data, Data Processing -Classification and Tabulation,
Cross Tabulation and its Significance.

Graphical Representation of Data - Line Graph, Bar Diagram, Pie Chart, Histograms,
Univariate Frequency Distribution - Measures of Central Tendency, Measures of Dispersion,
Skewness, Moments and Kurtosis, Numerical Problems.

Bivariate Frequency Distribution- Correlation, Various Methods, Regression Analysis,
Numerical Problems, Estimation of Population Parameters, Methods of Estimation,
Basic concepts of inference, Testing of hypothesis: t test, F test, Z test, Chi Square test; ANOVA
one way and Interpretation; ANOVA-two way and Interpretation; Numerical Problems;
Basics of Index Numbers.

Text/Reference Books:

1. Computational statistics. G. H. Givens and J. A. Hoeting, 2013.
2. Computational Statistics. James E. Gentle. 2009.
3. An introduction to statistical learning: with applications in R, G. James, D. Witten, T. Hastie, R. Tibshirani.
4. Wood SN (2015). Core Statistics. Cambridge University Press, New York.
5. Gentle JE, Härdle WK, Mori Y (2012). Handbook of Computational Statistics: Concepts and Methods. Second edition. Springer.

SEC-2-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 Template classes and their use.
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 (including assignment operators, unary operators)
Inheritance, Polymorphism and Exception Handling: Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions),

Abstract class, Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions.

SEC-2-OOP-b: Object Oriented Programming through C++ Lab

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.

Note: Student has to submitted a lab notebook properly signed by the concerned faculty member 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

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 Stroustrup, 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.