UNIVERSITY OF CALCUTTA

NotificationNo.CSR/34/19

It is notified for information of all concerned that the Syndicate in its meeting held on 08.08.2019 (vide Item No.55), subsequently confirmed by the Syndicate dated 27.08.2019 (Item No.01) approved new revised Syllabi of B.Sc. (Honours/General) in Computer Science under CBCS, under this University, as laid down in the accompanying pamphlet.

The above shall be effective from the Odd Semester Examinations, 2019.

Prof.(Dr.) Debasis Das
Registrar (Acting)

SENATE HOUSE
KOLKATA-700 073
The 20th September, 2019.

UNIVERSITY
OF
CALCUTTA

SYLLABUS
of
Bachelor of Science (B. Sc.)
(Honours)
in
Computer Science (CMSA)
Choice Based Credit System (CBCS)
2018
SEMESTER – I - VI

Semester - I				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -1	Theory	CMS-A-CC-1-1-TH	Digital Logic	4
	Practical	CMS-A-CC-1-1-P	Digital Circuits	2
Core Course -2	Theory	CMS-A-CC-1-2-TH	Programming Fundamentals using C	4
	Practical	CMS-A-CC-1-2-P	Programming in C	2

Semester - II				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -3	Theory	CMS-A-CC-2-3-TH	Data structure	4
	Practical	CMS-A-CC-2-3-P	Data structure using C	2
Core Course -4	Theory	CMS-A-CC-2-4-TH	Basic Electronic Devices and Circuits	4
	Practical	CMS-A-CC-2-4-P	Basic Electronic Devices and Circuits	2

Semester - III				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -5	Theory	CMS-A-CC-3-5-TH	Computer Organization & Architecture	4
	Practical	CMS-A-CC-3-5-P	Computer Organization Lab	2
Core Course -6	Theory	CMS-A-CC-3-6-TH	Computational Mathematics	4
	Practical	CMS-A-CC-3-6-P	Computational Mathematics Lab	2
Core Course -7	Theory	CMS-A-CC-3-7-TH	Operating Systems	4
	Practical	CMS-A-CC-3-7-P	Operating Systems Lab	2
Skill Enhancement Course (SEC-A) (Candidate has to opt any one from the under mentioned courses)				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
SEC-A-1	Theory	CMS-A-SEC-A-3-1-TH	Computer Graphics	2
SEC-A-2	Theory	CMS-A-SEC-A-3-2-TH	IoT (Internet of Things)	2

Semester - IV				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -8	Theory	CMS-A-CC-4-8-TH	Data communication, Networking and Internet technology.	4
	Practical	CMS-A-CC-4-8-P	Computer Networking and Web Design Lab.	2
Core Course -9	Theory	CMS-A-CC-4-9-TH	Introduction to Algorithms & its Application.	4
	Practical	CMS-A-CC-4-9-P	Algorithms Lab.	2
Core Course - 10	Theory	CMS-A-CC-4-10-P	Microprocessor and its Applications.	4
	Practical	CMS-A-CC-4-10-P	Programming with Microprocessor 8085.	2
Skill Enhancement Course (SEC-B) Candidate has to opt any one from the under mentioned courses)				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
SEC-B-1	Theory	CMS-A-SEC-B-4-1-TH	Information Security	2
SEC-B-2	Theory	CMS-A-SEC-B-4-2-TH	E-Commerce	2

Semester - V				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -11	Theory	CMS-A-CC-5-11-TH	Database Management system (DBMS)	4
	Practical	CMS-A-CC-5-11-P	RDBMS lab using My SQL & PHP	2
Core Course -12	Theory	CMS-A-CC-5-12-TH	Object Oriented Programming (OOPs)	4
	Practical	CMS-A-CC-5-12-P	OOPs Lab using JAVA	2
Semester -V (DSE) Discipline Specific Elective Course - DSE-A(1&2)& DSE-B(1&2), (Candidates have to opt one course from DSE-A & one course from DSE-B)				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
DSE-A-1	Theory	CMS-A-DSE-A-1-TH	Digital Image Processing	4
	Practical	CMS-A-DSE-A-1-P	Image processing Lab	2
DSE-A-2	Theory	CMS-A-DSE-A-2-TH	Data Mining & its Application	4
	Practical	CMS-A-DSE-A-2-P	Data Mining Lab	2
DSE-B-1	Theory	CMS-A-DSE-B-1-TH	Operation Research (O.R)	4
	Practical	CMS-A-DSE-B-1-P	Operation Research (O.R) Lab	2
DSE-B-1	Theory	CMS-A-DSE-B-2-TH	Programming using Python	4
	Practical	CMS-A-DSE-B-2-P	Programming in Python Lab	2

Semester - VI				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -13	Theory	CMS-A-CC-6-13-TH	Software Engineering	4
Core Course -14	Theory	CMS-A-CC-6-14-TH	Theory of Computation	4
	Practical	CMS-A-CC-6-14-P	Project	4
Semester - VI (DSE) Discipline Specific Elective Course - DSE-A(3&4)& DSE-B (3&4) (Candidates have to opt one course from DSE-A & one course from DSE-B)				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
DSE-A-3	Theory	CMS-A-DSE-A-3-TH	Embedded Systems	4
	Practical	CMS-A-DSE-A-3-P	Embedded Systems Lab	2
DSE-A-4	Theory	CMS-A-DSE-A-4-TH	Multimedia and its Application	4
	Practical	CMS-A-DSE-A-4-P	Multimedia and its Application Lab	2
DSE-B-3	Theory	CMS-A-DSE-B-3-TH	Introduction to Computational Intelligence	4
	Practical	CMS-A-DSE-B-3-P	Computational Intelligence Lab	2
DSE-B-4	Theory	CMS-A-DSE-B-4-TH	Advance Java	4
	Practical	CMS-A-DSE-B-4-P	Advance Java Lab	2

Semester - I				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -1	**Theory**	**CMS-A-CC-1-1-TH**	**Digital Logic**	**4**
	Practical	**CMS-A-CC-1-1-P**	**Digital Circuits**	**2**
Core Course -2	**Theory**	**CMS-A-CC-1-2-TH**	**Programming Fundamentals using C**	**4**
	Practical	**CMS-A-CC-1-2-P**	**Programming in C**	**2**

CMS-A-CC-1-1-TH: Digital Logic
Core Course-1: Theory, Credits-04, Contact hours - 60.

Introduction to Computer fundamentals Central Processing Unit (CPU), Primary and Secondary Storage devices, I/O Devices, Classification of Computers: Super, Mainframe, Mini and Personal Computer, System and Application Software.	**02 hours**
Number Systems Weighted and Non - Weighted Codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Gray Codes, Alphanumeric codes, ASCII, EBCDIC, Conversion of bases, 1's, 2's complement representation, Parity bits. **Single bit error detection and correcting codes:** Hamming Code. **Fixed and Floating Point Arithmetic:** Addition, Subtraction, Multiplication and Division.	**05 hours**
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 up to four (4) variables, Two level and multilevel implementation using logic gates, simplification of logic expressions.	**08 hours**
Combinational Circuits **Adder & Subtractor:-** Design and Construction of Half adders (2-bit) & Subtractor (2-bit), Full Adder (3-bit) & Subtractor (3-bit) using basic logic gates (OR, AND, NOT) and universal logic gates (NAND & NOR). **Multibit Adder:-** Ripple Carry Adder, Carry Look Ahead (CLA) Adder, BCD Adder, design & construct 1'S & 2'S Complement Adder/Subtractor unit using 4-bit full adder units, 1-bit, 2-bit, 3-bit and 4-bit magnitude comparator using basic logic gates. **Data Selector-Multiplexer:** Expansion (Cascading), function realization, Universal function realization, Multifunction realization. **Encoders:-** Realization of simple Encoders and priority Encoders using Basic and Universal Logic gates. **Data Distributor:-** De-multiplexer, Cascading, realization of various functions.	**20 hours**

Chip Selector/Minterm Generator - Decoder- Function Realization, BCD Decoders, Seven Segment Display and Decoders. **Parity bit and Code Converters:** Parity bit Generator/Checker, Gray to Binary code converter, Binary to Gray Code Converter.	
Sequential Circuits **Latch:** Set/Reset (SR) 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, Clock - Duty Cycle, rising time, falling time, negative and positive edge detector circuits, edge triggered SR, D and JK flip flop, flip-flop Conversions, flip-flops with preset/set and clear/reset asynchronous inputs. **Registers:** Serial Input Serial Output (SISO), Serial Input Parallel Output (SIPO), Parallel input Serial Output (PISO), Parallel Input Parallel Output (PIPO), Universal Shift Registers. **Counters:** Asynchronous Counter: UP/DOWN Counters, Mod - N Counters, BCD Counter (Counter Construction using J-K and T Flip Flops). **Synchronous Counter:** UP/DOWN Counters, Mod-N Counters, Ring & Johnson Counters.	**21 hours**
Integrated Circuits (Qualitative study only) **Bipolar Logic Families:** DTL, TTL NOT Gate, TTL NAND Gate, TTL NOR Gate, Open Collector, Fan-in, Fan-out. **MOS Logic Families:** NMOS, PMOS, CMOS, SSI, MSI, LSI and VLSI classification (concepts only).	**04 hours**

CMS-A-CC-1-1-P: Digital Circuits
Core Course-1: Practical, Credits - 02, Contact hours - 40.

Combinational Circuits

1. Implementation of different functions (SOP, POS) using basic (AND, OR and NOT) logic gates.
2. Study and prove De-Morgan's Theorem.
3. Realization of Universal functions using NAND and NOR gates.
4. Implementation of half (2-bit) and full adder (3-bit) using basic (AND, OR and NOT) and Universal logic gates (NAND & NOR).
5. Implementation of half (2-bit) and Full Subtractor (3-bit) using basic (AND, OR and NOT) and Universal logic gates (NAND & NOR).
6. Design and implement 1-Digit BCD adder using 7483/74283 and other necessary logic gates.
7. Design 4 to 1 multiplexer using basic or Universal logic gates and implement half and full adder/subtractor.
8. Design and implement half and full adder /subtractor and other functions using multiplexers 74151/74153 and other necessary logic gates.
9. Cascading of Multiplexers.
10. Design 2 to 4 decoder using basic or universal logic gates.
11. Study 74138 or 74139 and implement half and full Adder/Subtractor and other functions.

12. Implementation of 1-bit magnitude comparator using decoders (74138/74139) and other necessary logic gates.
13. Cascading of Decoders.
14. Study magnitude comparators 7485.
15. Design and construct magnitude comparator (2-bit) using basic (AND, OR & NOT) and universal (NAND/NOR) logic gates.
16. Design a display unit using Common anode or cathode seven segment display and decoders (7446/7447/7448)
17. Design and implement 4-input 3-output (one output as valid input indicator) priority encoder using basic (AND, OR & NOT) logic gates.
18. Study Priority Encoder IC 74147/74148.
19. Design a parity generator and checker using basic logic gates

Sequential Circuits

1. Realization of SR, D, JK Clocked/Gated, Level Triggered flip-flop using basic or Universal logic gates.
2. 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 n-bit Synchronous Counter, where n is maximum 4.
5. n-bit binary arbitrary sequence synchronous counter where n is maximum 4.

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, Morris Mano, PHI.
6. Digital Integrated Electronics, H.Taub & D.Shilling, McGraw Hill.
7. Digital Circuits and Design, Salivahan, Vikas.

CMS-A-CC-1-2-TH: Programming Fundamentals using C
Core Course-2: Theory: 04 Credits: 60 hours

Introduction: History, Basic Structure, Algorithms, Structured programming constructs.	04 hours
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.	08 hours
C Preprocessor: File inclusion, Macro substitution.	06 hours
Statements: Assignment, Control statements- if, if else, switch, break, continue, goto, Loops-while, do while, for.	06 hours

Functions: Argument passing, return statement, return values and their types, recursion	06 hours
Arrays: String handling with arrays, String handling functions.	07 hours
Pointers: Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation.	10 hours
User defined Data types: Enumerated data types, Structures. Structure arrays, Pointers to Functions and Structures, Unions	07 hours
File Access: Opening, Closing, I/O operations.	06 hours

CMS-A-CC-1-2-P: Programming with C
Core Course-2: Practical: 02 Credits: 40 hours

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. 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.

25. Add two distances in meter kilometer system using structure.

26. Add two complex numbers using structures.

27. Calculate the difference between two time periods using structures.

These are only examples; more can be included related to the theory.

Use open source C compiler.

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, Kanetkar, BPB Publication.
5. Programming in ANSI C, Balaguruswamy, McGraw Hill.
6. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

Semester - II				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -3	**Theory**	**CMS-A-CC-2-3-TH**	**Data structure**	**4**
	Practical	**CMS-A-CC-2-3-P**	**Data structure using C**	**2**
Core Course -4	**Theory**	**CMS-A-CC-2-4-TH**	**Basic Electronic Devices and Circuits**	**4**
	Practical	**CMS-A-CC-2-4-P**	**Basic Electronic Devices and Circuits**	**2**

CMS-A-CC-2-3-TH: Data Structure
Core Course-3: Theory, Credits - 04, Contact hours - 60.

Introduction to Data Structure Abstract Data Type.	**01 hour**
Arrays 1D, 2D and Multi-dimensional Arrays, Sparse Matrices. Polynomial representation	**05 hours**
Linked Lists Singly, Circular and Doubly Lists, Polynomial representation.	**09 hours**
Stacks Array and linked representation of stack, Prefix, Infix and Postfix expressions, utility and conversion of these expressions from one to another, evaluation of postfix and prefix expression using stack, applications of stack, limitations of Array representation of stack.	**05 hours**
Queues Array and Linked representation of Queue, Circular Queue, De-queue, Priority Queues.	**05 hours**
Recursion Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation), Tail recursion.	**05 hours**
Trees Introduction to Tree as a data structure: Binary Trees (Recursive and Iterative Traversals), Binary Search Tree (Traversal, Insertion, Deletion and Searching), Threaded Binary Trees (Traversal and advantages).	**15 hours**
Searching and Sorting Linear Search, Binary Search, Comparison of Linear and Binary Search with respect to time complexity, Selection Sort, Bubble sort, Insertion Sort, Merge Sort, Quick sort, Heap sort, Shell Sort, Radix sort, Comparison of Sorting Techniques with respect to time complexity.	**10 hours**
Hashing Introduction to Hashing, Different hashing Techniques, Collision and resolving collision by Open Addressing, Closed Hashing, Separate Chaining, Choosing a Hash Function.	**05 hours**

CMS-A-CC-2-3-P: Data Structure Lab using C.
Core Course- 3: Practical, Credits - 02, Contact hours - 40.

Lab based on Data Structure theory except Threaded Binary Tree, Shell Sort, Radix Sort and hashing

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search.
2. Write a program to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement 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 Queue operations using Array and linked list implementation.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. Write a program to scan a polynomial using linked list and add two polynomials.
11. Write a program to create a Binary Search Tree and include following operations in tree:
    - **(a)** Insertion (Recursive and Iterative Implementation).
    - **(b)** Deletion.
    - **(c)** Search a node in BST.
    - **(d)** Display its preorder, postorder and inorder traversals recursively.
    - **(e)** Display its preorder, postorder and inorder traversals Iteratively.
    - **(f)** Display its level-by-level traversals.
    - **(g)** Count the non-leaf nodes and leaf nodes.
    - **(h)** Display height of tree.
    - **(i)** Create a mirror image of tree.
12. Write a program to reverse the order of the elements in the stack using additional stack.
13. Write a program to reverse the order of the elements in the stack using additional Queue.

Note: These are only sample programs; more can be included related to the theory.

Text/ Reference Books

1. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, University Press.
2. Data Structures: A Pseudocode Approach with C, Richard F. Gilberg and Behrouz A. Forouzan, Cengage Learning.
3. Data Structure using C, E Balagurusamy, McGraw Hill.
4. Data Structures Using C and C++, Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, PHI.
5. Classic Data Structures, Debasis Samanta, Second Edition, EEE, PHI.
6. Data Structures, Seymour Lipschutz, Schaum's Outlines, Tata McGraw Hill.
7. Data Structures Through C (A practical approach), G.S Baluja, DhanpatRai& Co.

CMS-A-CC-2-4-TH: Basic Electronic Devices and Circuits
Core Course-4: Theory, Credits - 04, Contact hours - 60.

Basics of Circuit Theory KVL, KCL, Thevenin's, Norton's, superposition, maximum power transfer theorem, application to simple problems.	**04 hours**
Theory of Semiconductor devices Semiconductor materials and their properties, classification based on energy band diagram, Intrinsic and extrinsic semiconductors, P & N type.	**03 hours**
Diode and its applications **PN junction diode:** Construction, characteristics and working principle, unbiased and biased band diagram, Single Phase Half and Full wave rectifier circuits, working principle, derivation and calculation of average dc current, average dc voltage, RMS, ripple factor, efficiency, Peak Inverse Voltage (PIV), Circuit and working of bridge rectifiers. **Zener diode:** Characteristics and its application as a voltage regulator, simple problems.	**09 hours**
Bipolar Junction Transistor Working Principle of Junction bipolar Transistor (including current components, current gains), **Modes:** Common Emitter (CE), Common Base (CB), Common Collector (CC), **DC biasing in CE mode:** Fixed bias, Emitter Stabilized bias, Voltage divider bias and Collector feedback bias, simple related numerical problems, Q-Point, dc load line analysis, single stage CE mode based transistor amplifying action (qualitative study). **Inverter using transistors:** Transistor as a switch, transfer characteristics and threshold voltages.	**08 hours**
Unipolar Junction Transistor Principle of JFET and MOSFET, Depletion and Enhancement mode operations, Concept of NMOS, PMOS and CMOS. CMOS circuits for basic logic gates (AND, OR, NOT, NAND and NOR).	**08 hours**
PNPN Devices **Construction, characteristics, working and simple applications:** SCR, DIAC, TRIAC. **Power supply (qualitative study only):** SCR regulated power supply, Switch Mode Power Supply (SMPS).	**06 hours**
Optoelectronic materials (Qualitative study) **Construction and working**: LED, LCD, Photo Sensors and basics of Optical fiber and Opto-couplers).	**02 hours**
Operational Amplifiers (OPAMP) Ideal Characteristics, Open loop operation, Single and double ended operation, Common mode operation, Common mode rejection ratio (CMRR), Offset parameters, Concept of Virtual ground. **Application:** Inverting, Non-inverting Amplifier, Inverting and Non-inverting Adder, Differentiator, Integrator, Scale changer and Schmitt Trigger. **Signal Generation using OPAMP:** Monostable, Astable (Square wave generator).	**12 hours**
Timer Construction and Functional description of 555, Mono-stable, Bi-stable and Astable Operation, VCO.	**04 hours**

Data Acquisition **Digital to Analog Converter (DAC):** R-2R ladder, Weighted resistor type. **Analog to Digital Converters (ADC):** Flash, Counter, Successive Approximation Register (SAR), Dual Slope type.	**04 hours**

CMS-A-CC-2-4-P: Basic Electronic Devices and Circuits Lab.
Core Course-4: Practical, Credits - 02, Contact hours - 40.

1. Study the forward characteristic of a p-n junction diode and calculate the static and dynamic resistance.
2. Construct a Half wave rectifier using power diodes and study its load regulation characteristics with and without capacitor filter.
3. Construct a full wave rectifier using power diodes and study its load regulation characteristics with and without capacitor filter.
4. Construct a Bridge rectifier using power diodes and study its load regulation characteristics with and without capacitor filter.
5. Study the reverse characteristic of a Zener diode and calculate the Zener voltage from the characteristic curve and also calculate the value of current limiting resistance.
6. Construct a voltage regulator using Zener diode and study its load regulation characteristics.
7. Construct a positive and negative voltage regulator using three terminal linear voltage regulators 78XX and 79XX. Study its load regulation characteristics.
8. Construct a variable positive voltage regulator using three terminal linear voltage regulator LM317 and study its load regulation characteristics for different sets of output voltage.
9. Study the output characteristics of a transistor in CE mode and calculate the dc current gain (β) from the graph.
10. Realize a NOT operation using a Transistor. Draw its transfer characteristics and measure the threshold voltage.
11. Construct and study an Inverting amplifier using OPAMP with different sets of voltage gain and calculate the gain from the graph.
12. Construct and study a non-inverting amplifier using OPAMP with different sets of voltage gain and calculate the gain from the graph.
13. Construct and study an inverting adder using OPAMP capable of adding two inputs.
14. Construct and study a non-inverting adder using OPAMP capable of adding two inputs.
15. Construct and study a subtractor using OPAMP.
16. Construct and study the OPAMP as a subtractor.
17. Construct and study the OPAMP as a differentiator. Apply sine and square wave and study and record the output waveforms.
18. Construct and study the OPAMP as an integrator. Apply sine and square wave and study and record the output waveforms.
19. Construct an astable multivibrator using Timer 555.
20. Construct and study a R-2R ladder digital to analog converter.
21. Convert an analog signal into digital using ADC 0804/0808/0809.

Text/Reference Books

1. Electronic Devices & Circuits Theory, Boylestad & Nashelsky, Prentice Hall.
2. Electronics fundamental & Application, Chattopadhyay, Rakshit, New Age International Publishers.
3. Op-Amps and Linear Integrated Circuits, R. A. Gayakwad, Prentice Hall.

4. Linear Integrated Circuits, D. Roy Choudhury, Shail B. Jain, fourth edition, New Age International Publishers.
5. Solid State Electronic Devices, Streetman, PHI.
6. Elements of Electronics, Bagde Singh, S Chand Publication.
7. Microelectronic circuits, Sedra Smith, Oxford.
8. Operational Amplifier and Linear Integrated Circuits, Coughlin Driscol.
9. Electronic Devices and Circuits, Salivahanan, Suresh Kumar, McGraw Hill education.

Semester - III				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -5	**Theory**	**CMS-A-CC-3-5-TH**	**Computer Organization & Architecture**	**4**
	Practical	**CMS-A-CC-3-5-P**	**Computer Organization Lab**	**2**
Core Course -6	**Theory**	**CMS-A-CC-3-6-TH**	**Computational Mathematics**	**4**
	Practical	**CMS-A-CC-3-6-P**	**Computational Mathematics Lab**	**2**
Core Course -7	**Theory**	**CMS-A-CC-3-7-TH**	**Operating Systems**	**4**
	Practical	**CMS-A-CC-3-7-P**	**Operating Systems Lab**	**2**
Skill Enhancement Course (SEC-A) (Candidate has to opt any one from the under mentioned courses)				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
SEC-A-1	**Theory**	**CMS-A-SEC-A-3-1-TH**	**Computer Graphics**	**2**
SEC-A-2	**Theory**	**CMS-A-SEC-A-3-2-TH**	**IoT (Internet of Things)**	**2**

CMS-A-CC-3-5-TH: Computer Organization and Architecture
Core Course- 5: Theory, Credits:04, Contact hours: 60.

Basic Structure of Computers (Qualitative Discussion) Computer Types, Basic Functional Units, Basic Operational Concept, Bus Structure, Software, Performance, Multiprocessor and Multicomputer, IAS Computer, Historical perspectives.	**05 hours**
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.	**05 hours**
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.	**05 hours**
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, Control Word, Accumulator Based, Register Based, Stack Type CPU organization.	**06 hours**
Control Unit Hardwired Control Unit, Micro-programmed Control Unit: Control memory, Address Sequencing, conditional branching, mapping of instructions, subroutine, Design of Control Unit.	**07 hours**
CPU Registers Program Counter, Stack Pointer Register, Memory Address Register, Instruction Register, Memory Buffer Register, Flag registers, Temporary Registers.	**06 hours**
Instructions. Operational Code, Operands, Zero, One, Two and Three Address Instruction, Instruction Types, Addressing modes, Data Transfer and Manipulation instructions, Program control instructions.	**03 hours**

CISC and RISC processors Introduction, relative merits and De-merits.	**03 hours**
Computer Peripherals VDU, Keyboard, Mouse, Printer, Scanner (Qualitative approach).	**08 hours**
Input / Output Organization Polling, Interrupts, subroutines, Memory mapped IO, IO mapped IO, DMA, I/O Bus and Protocol, SCSI, PCI, USB, Bus Arbitration.	**02 hours**
Memory Primary memory: ROM, PROM, EPROM, EEPROM, Flash memory, RAM: SRAM, DRAM, Asynchronous DRAMs, Synchronous DRAMs, Structure of Larger Memories, RAMBUS Memory, Cache Memory: Mapping Functions, Replacement Algorithms, interleaving, Hit and Rate penalty, Virtual memories, Address Translation, Memory Management requirements, Secondary Storage: Magnetic Hard Disks, Optical Disks, Magnetic Tape Systems.	**10 hours**

CMS-A-CC-3-5-P: Computer Organization Lab.
Core Course-5, Practical, Credits: 02, Contact hours:40.

(1). Construct an Arithmetic Unit capable of performing 4-bit subtraction and Addition using 2's complement method. Use Parallel Adders and other necessary logic gates.
(2). Construct a logical unit using logic gates capable of performing 4-bit, Bitwise ORing, ANDing, XORing and inversion.
(3). Construct a 4-bit ALU unit which can perform the following operation;

Selection		Function
S_1	S_0	
0	0	Addition
0	1	Subtraction
1	0	XOR-ing
1	1	Complement

(4). Construct a 2-bit Carry Look Ahead Adder using logic gates.
(5). Study and Construct a 1-digit BCD/Decimal adder using parallel adders and other necessary logic gates.
(6). Construct a Binary Multiplier using basic logic gates.
(7). Construct a Binary Divider using basic logic gates.
(8). Subtraction with 1's complement method using parallel adders and other necessary logic gates.
(9). Construction of BCD Subtractor with 9'S complement method using parallel adders and logic gates.
(10). Construction of BCD Subtractor with 10'S complement method using parallel adders and logic gates.
(11). Binary magnitude comparators (up to 4 bits) using parallel adder and logic gates.
(12). Construct a Binary 4-bit and 8-bit adder using logic gates.
(13). Construct a Serial in Serial out 4-bit register.
(14). Construct a 4-bit Universal Shift register.
(15). Construct a 4 bit ring counter.

(16). Construct a 4 - bit Johnson Counter.
(17) Construct RAM (4-bit) and extend it
(18). Horizontal and Vertical Cascading of Memory modules.
(19). Code converters using memory modules.

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.

CMS-A-CC-3-6-TH: Computational Mathematics
Core Course- 6: Theory, Credits: 04, Contact hours: 60.

Introduction Set Theory: Finite and Infinite Sets, Uncountable Infinite Sets, Relations: Properties of Binary Relations, Closure, Partial Ordering Relations, Equivalence, Functions: definition, one-to-one, onto and invertible, Mathematical Functions: Exponential and Logarithmic, Counting: Mathematical Induction, Pigeonhole Principle, Permutation and Combination, Binomial Theorem, Principle of Inclusion and Exclusion.	**10 hours**
Introduction to Probability Elementary events, Sample space, Classical and Axiomatic definition of Probability, Theorems on Total Probability, Conditional Probability, Bernoulli Trials and Binomial Distribution, Bayes' Theorem, Random Variables, Expectation, Variance, Standard Deviation.	**10 hours**
Growth of Functions Asymptotic Notations, Standard notations and common functions with simple examples.	**04 hours**
Recurrences Relations, Generating Functions, Linear Recurrence Relations with Constant Coefficients and their solution, Substitution Method, Recurrence Trees.	**06 hours**
Numerical Methods (Algorithmic Approach) **Errors:** Approximate and Rounding of Numbers, Significant digits, Errors and their types, Propagation of errors. **Interpolation:** Newton Forward and Backward interpolation, Lagrange interpolation. Solving a Set of Linear Equations: Gaussian Elimination, Gauss–Jordan, Iteration methods and their convergence conditions, Gauss-Seidel, Gauss-Jacobi Iterative Methods. **Solving Non-linear equations:** Bisection, Regula-falsi, Secant and Newton-Raphson, their order of convergence. **Solving Differential Equations:** Euler, Runge-Kutta second and fourth order methods. **Numerical Integration:** Trapezoidal and Simpson's $1/3^{rd}$ rules. **Curve fitting :** Least square approximation, Linear regression, Polynomial regression, Fitting Exponential and Trigonometric functions.	**20 hours**

Graph Theory Basic Terminology, Models and Types, Multi graphs and Weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Trees and their basic terminologies and properties.	**10 hours**

CMS-A-CC-3-6-P: Computational Mathematics Lab.
Core Course- 6: Practical, Credits:02, Contact hours: 40.

Lab. based on Numerical Methods using C.

Text/ Reference Books:

1. Elements of Discrete mathematics, C.L. Liu & Mahapatra, Tata McGraw Hill.
2. Discrete Mathematics and Its Applications, Rosen, McGraw Hill.
3. Introduction to algorithms, T.H. Cormen, C.E. Leiserson, R. L. Rivest, Prentice Hall .
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. Numerical Analysis and Computational Procedures by Mollah, New Central Book.
8. Computer Oriented Numerical Methods, 3rd Edition, V Rajaraman, PHI
9. Graph Theory With Applications To Engineering And Computer Science by Narsingh Deo, PHI.
10. Graph Theory by J.A. Bondy and U.S.R. Murty, Springer.
11. Introduction to Graph Theory by D B West, 2nd edition, Pearson Education

CMS-A-CC-3-7-TH: Operating Systems
Core Course- 7: Theory, Credit: 04, Contact hours: 60.

Introduction Basic OS functions, types of operating systems- batch processing, multiprogramming, time sharing, multiprocessing, distributed and real time systems.	**6 hours**
Operating System Organization Processor and user modes, kernels, system calls and system programs.	**6 hours**
Process System view of the process and resources, process control block, I/O and CPU bound process, process hierarchy, concept of threads **Process Scheduling:** Preemptive and non-preemptive scheduling, Long term scheduling, short term/CPU scheduling (FCFS, SJF, SRJF, RR and priority) and medium term scheduling **Process Synchronization:** Concurrent processes, critical section, semaphores and application, methods for inter-process communication;	**18 hours**
Deadlock: Definition, Prevention, Avoidance, Detection, Recovery.	**9 hours**

Memory Management Physical and logical address space; memory allocation strategies – fixed and variable partitions, paging, segmentation, virtual memory	**14 hours**
File and I/O Management Directory structure, file operations, file allocation methods, disk management.	**5 hours**
Protection and Security Policy mechanism, Authentication	**2 hours**

CMS-A-CC-3-7-P: Operating Systems Lab.

Core Course- 7: Practical, Credit: 02, Contact hours: 40.

Shell programming in LINUX

1. Write a shell script to convert the content of a file from lower case to upper case.
2. Write a shell script to count the words, lines and characters of a given file. File name should be provided at run time.
3. Write a shell script that take a word from user and find out the frequency of the word in a given file.
4. Write a shell script that gets executed at the moment of user login and it displays Good Morning, Good afternoon, Good Evening, Good Night, depending upon the time at which the user logs on.
5. Write a shell script to print Pascal diamond.
6. Write a shell script to find a number using sequential search method.
7. Write a shell script to find a number using binary search technique.
8. Write a shell script to sort a set of integer numbers using bubble sort.
9. Write a shell script to find out the factorial of a given number.
10. Write a shell script to reverse a string and check whether it is a palindrome.
11. Write a shell script to find the roots of a quadratic equation $ax^2 + bx + c = 0$, considering all possible cases.
12. Write a shell script for menu based system to insert records for employees with employee ID, name, designation, salary in a data file, also display records when necessary. Display salary for the employee asked.

These are only examples, more can be included.

Text/ Reference Books

1. Operating Systems Concepts, A Silberschatz, P.B. Galvin, G. Gagne, John Wiley Publications.
2. Modern Operating Systems, A.S. Tanenbaum, 3rd Edition, 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. Sumitabha Das , UNIX Concepts and Applications, Tata McGraw-Hill.
7. Understanding the Linux Kernel,D. P. Bovet and M. Cesati, O'Reilly.

CMS-A-SEC-A-3-1-TH: Computer Graphics

Skill Enhancement Course: SEC-A: Choice -1: Theory, Credit:02, Contact hours: 40.

Introduction Basic concepts of Graphics Devices– CRT monitor, Monochrome and Color Monitor displaying technique only, Physical and logical units of graphics devices – Pixel and its different properties, Basic idea for image or picture formation using pixels – Raster Scan and Vector Scan.	**05 hours**
Basic geometrical shapes formation algorithms Concepts Co-ordinate System, Line Segment, Digital Differential Analyzer, Circle and arc segment, elliptic segment, Bresenham's and Midpoint scan conversion algorithms.	**05 hours**
Two and Three Dimensional Transformations Transformations operations - Translation, Rotation, Scaling. Reflection, Shearing and Inverse of these operations, Homogeneous coordinate system representation, matrix representation. Composite Transformations Operations – Basic ideas and matrix representations by matrix concatenation for a particular operation.	**14 hours**
Two Dimensional Clipping View port, window port, display device, Point Clipping, Line Clipping, Cohen-Sutherland line clipping algorithm, Sutherland Hudgeman polygon clipping algorithm	**08 hours**
Projection Basic Concept of Projection operation and its application, Classification – Perspective, Parallel and its subclasses, Principles of these projections (Geometric representation only, no Mathematical Foundation and algorithms)	**06 hours**
Applications Basic Concepts Computer Art, Animation – Animating and modeling of real world, Morphing – Classification of morphing and Application to the Advertisements and publicities.	**02 hours**

Text/ Reference Books:

1. Computer Graphics by Zhigang Xiang, Roy Plastock, Schaum's Outlines Series.
2. Computer Graphics by Hearn & Baker, Pearson.
3. Procedural Elements for Computer Graphics by David F. Roger, 2nd Edition, TMH.
4. Computer Graphics by Foley, Van Dam, Feimer& John, Pearson.
5. Introduction to Computer Graphics and Multimedia, Mukhophadhyay and Chattopadhyay, Vikas publication.

CMS-A-SEC-A-3-2-TH: Internet of Things (IoT)

Skill Enhancement Course: SEC-A: Choice -2, Theory, Credit:02, Contact hours: 40.

Introduction to Internet of Things (IoT) Defining IoT, Characteristics of IoT, Physical design of IoT, Functional blocks of IoT, Communication models & APIs.	**04 hours**
IoT and M2M Difference between IoT and M2M, Software defined Network, network function virtualization (NFV), difference between SDN and NFV.	**04 hours**

Network & Communication aspects Wireless medium access issues, MAC protocol survey, Survey routing protocols, Sensor deployment & Node discovery, Edge connectivity and protocols, Fog/Gateway Devices for Data aggregation and dissemination, Security challenges.	**08 hours**
IoT Physical Servers and Cloud Offerings Introduction to Cloud Storage models and communication APIs Web Server – Web Server for IoT, Cloud for IoT, Python web application framework.	**05 hours**
Developing IoTs Introduction to Python, Introduction to different IoT tools, Developing applications through IoT tools, Developing sensor based application through embedded system platform, Implementing IoT concepts with python.	**08 hours**
IoT Physical Devices and Endpoints Introduction to Raspberry PI-Interfaces (serial, SPI, I2C) Programming – Python program with Raspberry PI with focus of interfacing external gadgets.	**04 hours**
IoT Analytics Signal processing, real-time and local analytics, Databases, cloud analytics and applications.	**04 hours**
Domain specific applications of IoT Home automation, Industry applications, Surveillance applications.	**03 hours**

Text/ Reference Books:

1. Foundational Elements of an IoT Solution,J. Biron and J. Follett, O'Reilly Media.
2. IoT fundamentals, David, Pearson Education.
3. Internet of Things by Tripathy and Anuradha, CRC Press.
4. Internet of Things – A hands-on approach, ArshdeepBahga and VijayMadisetti, Universities Press.
5. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly.
6. Internet of Things by Bahga, Madishetty, Orient Blackswanpvtltd.

Semester - IV				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -8	**Theory**	**CMS-A-CC-4-8-TH**	**Data communication, Networking and Internet technology.**	**4**
	Practical	**CMS-A-CC-4-8-P**	**Computer Networking and Web Design Lab.**	**2**
Core Course -9	**Theory**	**CMS-A-CC-4-9-TH**	**Introduction to Algorithms & its Application.**	**4**
	Practical	**CMS-A-CC-4-9-P**	**Algorithms Lab.**	**2**
Core Course - 10	**Theory**	**CMS-A-CC-4-10-P**	**Microprocessor and its Applications.**	**4**
	Practical	**CMS-A-CC-4-10-P**	**Programming with Microprocessor 8085.**	**2**
Skill Enhancement Course (SEC-B) Candidate has to opt any one from the under mentioned courses)				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
SEC-B-1	**Theory**	**CMS-A-SEC-B-4-1-TH**	**Information Security**	**2**
SEC-B-2	**Theory**	**CMS-A-SEC-B-4-2-TH**	**E-Commerce**	**2**

CMS-A-CC-4-8-TH: Data Communication, Networking and Internet Technology.
Core Course- 8: Theory, Credit: 04, Contact hours: 60.

Overview of Data Communication and Networking **Introduction:** Data communications Components, data representation, direction of data flow (simplex, half duplex, full duplex). **Network Hardware**: Physical structure (type of connection, topology), categories of network (LAN, MAN, WAN). **Internet:** Brief history, Protocols and standards, **Reference models:** OSI reference model, properties of all the layers, TCP/IP reference model, their comparative study.	**04hours**
Physical Layer **Data & Signals:** Analog & Digital Data and Signals, periodic and non-periodic signals, composite signals, bandwidth, bit rate, transmission of digital signals. **Transmission Impairments:** Attenuation, Distortion and Noise. **Data Rate Limits:** Noiseless Channel: Nyquist Data rate, Noisy Channel: Shannon's Capacity, calculation of data rate using both limits. **Digital Transmission** **Digital to Digital Conversion**: Line coding, schemes (RZ, NRZ, Manchester, Differential Manchester), block coding. **Analog to Digital Conversion**: Sampling, Nyquist rate of sampling, Pulse code modulation (PCM), Delta Modulation (DM), Adaptive Delta Modulation (ADM), parallel and serial transmission. **Analog Transmission** **Digital to Analog:** Amplitude shift keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK), Quadrature Amplitude Modulation (QAM). **Analog to Analog Conversion:** Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation.	**12hours**

Bandwidth Utilization Techniques **Multiplexing:** FDM, Synchronous & Statistical TDM, WDM.	**04hours**
Transmission Medium **Guided media:** Twisted pair, Coaxial, Fiber optics. **Unguided:** Radio waves, microwaves, Infrared, Antenna, Communication satellites (qualitative study only).	**06hours**
Switching and Telephone network Circuit switched networks, Packet Switched networks, Virtual Circuit switch. Major components of telephone network, Dial up modem, DSL and ADSL modems, Cable TV for data transfer (qualitative study only)	**04hours**
Data link Layer: Types of errors, framing (character and bit stuffing), error detection & correction methods, Linear and cyclic codes, checksum. **Protocols:** Stop & wait ARQ, Go-Back- N ARQ, Selective repeat ARQ, HDLC (qualitative study only). **Physical addressing:** MAC address and its format.	**04hours**
Medium Access sub layer Point to Point Protocol, Token Ring: Reservation, Polling. **Multiple access protocols:** Pure & Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA. **Channelization:** FDMA, TDMA, CDMA (Qualitative study only). **Wired and Wireless LAN:** Standards, fast Ethernet, Protocol 802.11, Bluetooth.	**08hours**
Network layer **Internetworking & devices:** Repeaters, Hubs, Bridges, Switches, Router, Gateway, **Addressing:** IP addressing, Subnetting, **Routing techniques:** static vs. dynamic routing , **Protocols:** RARP, ARP, IP, ICMP	**11 hours**
Transport layer **Process** to Process delivery: UDP, TCP	**03 hours**
Application Layer Introduction to DNS, Remote logging, FTP, Electronic mail, WWW & HTTP	**04hours**

CMS-A-CC-4-8-P: Computer Networking and Web Design Lab

Core Course- 8: Practical, Credit: 02, Contact hour: 40.

Computer Networks: Practical Familiarization with Networking cables (CAT5, CAT6, UTP), Connectors (RJ-45, T-connector), Hubs, Switches, LAN installation & configuration (peer-to-peer) process.	**05 hours**
Web Design: Practical Web page design by HTML	
Handling HTML form HTML Capturing Form Data, GET and POST form methods, Dealing with multi value fields Redirecting a form after submission.	**20 hours**
Array Anatomy of an Array ,Creating index based and Associative array, Accessing array Looping with Index based array, with associative array using each() and for each() Some useful Library function.	**15 hours**

Text/ Reference Books:

1. Data Communication and Networking, B.A. Forouzan, Tata McGraw Hill.
2. Computer Networks, A.S. Tanenbaum, Pearson Education .
3. Data and Computer Communication, W. Stallings, Pearson Education.
4. Data & Computer Communication,Black, PHI.
5. Internet & World Wide Web: How to program,Harvey M. Deitel& Paul J. Deitel.
6. CGI Programming on the world wide web,ShishirGundavaram, O'Relly and Associates.

CMS-A-CC-4-9-TH: Introduction to Algorithms & its Applications
Core Course- 9: Theory, Credit: 04, Contact hours: 60.

Introduction to Algorithms: Definition, Characteristics, Recursive and Non-recursive algorithms.	**05 hours**
Asymptotic Complexity Analysis of Algorithms: Space and Time Complexity, Efficiency of an algorithm, Growth of Functions, Polynomial and Exponential Complexity, Asymptotic Notations: Big O Notation and Small o notation, Big Ω and Small ω, Big Θ and Small φ Notations, Properties: Best case/worst case/average case analysis of well-known algorithms.	**10 hours**
Algorithm Design Techniques: Concepts and simple case studies of Greedy algorithms. Divide and conquer: Basic concepts, Case study of selected searching and sorting problems using divide and conquer techniques: Dynamic programming: General issues in Dynamic Programming.	**15 hours**
Graph Representation and Algorithm: Graph traversal algorithms: BFS, DFS, Minimal spanning trees: Prim's Algorithm, Kruskal's Algorithm, Shortest path algorithms: Floyd's Algorithm, Floyd-Warshall Algorithm, Dijkstra's Algorithm, Graph Coloring Algorithms.	**25 hours**
Classification of Problems: Concept of P, NP.	**05 hours**

CMS-A-CC-4-9-P: Algorithms Lab.
Core Course- 9: Practical, Credit:02, Contact hour: 40.

Lab. based on Graph Theory using C **Graph Algorithms:** Implementation of Graph algorithms: Single Spanning Tree Generation using - BFS, DFS, Minimal Spanning Tree Generation using - Prim's Algorithm, Kruskal's Algorithm, Shortest Path finding using - Floyd's Algorithm, Floyd-Warshall Algorithm, Dijkstra's Algorithm, Graph Partitioning Algorithm.

Text/References Books:

1. Introduction to Algorithms, Cormen, Leiserson, Rivest and Stein, TMH.
2. The Design and Analysis of Algorithms, Aho, Hopcroft and Ullman, Pearson Education.
3. The Art of Computer Programming, D.E. Knuth, Pearson Education.
4. Algorithm Design, Jon Kleiberg and Eva Tardos,Pearson Education.
5. Data Structures and Algorithms - K.Mehlhorn.

6. Computer Algorithms, S.Baase, Pearson Education.
7. Fundamentals of Computer Algorithms, E. Horowitz and Sahani, Galgotia
8. Combinational Algorithms- Theory and Practice, E.M. Reingold, J. Nievergelt and N. Deo, PHI.

CMS-A-CC-4-10-TH: Microprocessor and its Applications
Core Course- 7: Theory, Credits:04, Contact hours: 60.

Introduction to Microcomputer based system: Evolution of Microprocessor and Microcontrollers and their advantages and disadvantages.	**03 hours**
Microprocessor Architecture and Memory Interfacing Basic Architecture of Microprocessor 8085 and explanation of each block, Microprocessor 8085 pin out and signals, Addressing modes, Instruction Formats, Instruction Cycle, Clock Cycle, Multiplexed Address Data Bus, Control and Status signals, Microprocessor and Bus Timing, De-multiplexing of Address Data Bus, Generation of Control Signals for I/O and Memory, Basic concepts in Memory Interfacing, Address Decoding and memory Addresses.	**14 hours**
Interfacing I/O Devices Basic Interfacing concepts, Peripheral I/O instructions (I/O mapped I/O), Device Selection and data Transfer, Absolute and Partial Decoding, Input Interfacing, Interfacing I/O using decoders, Memory mapped I/O techniques, Data transfer schemes**,** Interfacing 8155 memory segment.	**10 hours**
Programming 8085 Instruction Set of 8085, Different Programming Techniques, Stack and Subroutines, Counter and Time Delays, Code Conversion, BCD Arithmetic and 16 bit Data Operation.	**10 hours**
Interfacing Peripheral Devices and Applications Interrupts: 8085 Interrupt, RST instructions, Software and Hardware interrupt, multiple Interrupts and Priorities, 8085 Vectored Interrupts, Restart as Software Instructions. Interfacing Digital to Analog Converters, Analog to Digital Interfacing, keyboard interfacing, interfacing 8255 (Mode - 0, BSR), Support IC chips- 8237/8257,8259	**13 hours**
Microprocessor 8086 The 8086 microprocessor- Architecture, Instruction set, Addressing modes, Interrupts, Memory interfacing with 8086.	**10 hours**

CMS-A-CC-4-10-P:Programming with Microprocessor 8085
Core Course- 10: Practical, Credits:02, Contact hours: 40.

1. Assembly Language Programming for Arithmetic Operations like Addition, Subtraction, Multiplication and Division on 8, 16 bit data.
2. Assembly Language Programming for different logical operations.
3. Assembly Language Programming for code conversions.
4. Assembly Language Programming for different sorting techniques.
5. Assembly Language Programming for memory block transfer.
6. Assembly Language Programming for AP series and Fibonacci series.
7. Assembly Language Programming for HCF, LCM etc.

8. Assembly Language Programming for Searching.
9. Assembly Language Programming for frequency distribution.
10. Block Replacement and transfer

Many more programs can be included related to the programming techniques of Microprocessor 8085

Text/Reference books

1. Microprocessor architecture, programming and applications with 8085/8085A, Ramesh Gaonkar, Penram International Publication (PRI).
2. Fundamental of Microprocessors and Microcontrollers, B.Ram, Dhanpat Rai Publications.
3. Microprocessors and Microcontrollers, Senthil, Saravanan, Jeevananthan, Oxford.
4. Advanced Microprocessors and Peripherals by Ray and Bhurchandi – McGrawHill.
5. Intel Corp. Micro Controller Handbook – Intel Publications.
6. Microprocessors and Interfacing Programming and hardware by Douglas V. Hall, McGraw Hill.
7. Microprocessor 8085 and its Interfacing, Mathur, PHI..
8. The Intel Microprocessors: 8086/8088, 80186, 80286, 80386 & 80486, Bary B. Brey, Prentice Hall, India.

Skill Enhancement Course: SEC-B: Information Security/ E-Commerce
CMS-A-SEC-B-4-1-TH: Information Security
Skill Enhancement Course: SEC-B: Choice-1: Theory, Credit:02, Contact Hours: 40.

Overview Overview of Security Parameters: Confidentiality, Integrity and availability-security violation, Assumptions and Trust- Security assurance, OSI security architecture.	**05 hours**
Cryptography Mathematical Tools for Cryptography, Symmetric Encryption Algorithm, Theory of Block cipher design, Symmetric cipher model, Risk assessment, quantitative and qualitative approaches, Network security management, Firewalls, Web and wireless security management, Computer security log management, IT security infrastructure, Operating system security, user security, program security.	**10 hours**
Finite Field and Number Theory Groups, Rings, Fields-Modular, Prime numbers, Fermat's and Euler's Theorem, Chinese remainder Theorem, Discrete Logarithm.	**03 hours**
Hash Functions and Digital Signatures Authentication requirement – Authentication function -MAC, Hash functions, Security of hash function, Hashing Algorithms: MD5.	**05 hours**
Internet Firewalls for Trusted System Roles of Firewalls, Firewall related terminology, Types of Firewalls, Firewall designs.	**02 hours**
E-Mail, IP & Web Security (Qualitative study) **E-mail Security:** Security Services for E-mail-attacks possible through E-mail, Pretty Good S/MIME. **IP Security**: Overview of IPSec, IP Security Architecture, Authentication Header, Encapsulation Security Payload. **Web Security:** Secure Socket Layer/Transport Layer Security, Basic Protocol, SSL	**05 hours**

Attacks, Secure Electronic Transaction (SET).	
Cyber **Cyber laws to be covered as per IT 2008** Definitions, Digital Signature And Electronic Signature. 1) [Section 43] Penalty and Compensation for damage to computer, computer system, etc. 2) [Section 65] Tampering with Computer Source Documents. 3) [Section 66 A] Punishment for sending offensive messages through communication service, etc. 4) [Section 66 B] Punishments for dishonestly receiving stolen computer resource or communication device. 5) [Section 66C] Punishment for identity theft. 6) [Section 66D] Punishment for cheating by personation by using computer resource. 7) [Section 66E] Punishment for violation of privacy. 8) [Section 66F] Punishment for cyber terrorism. 9) [Section 67] Punishment for publishing or transmitting obscene material in electronic form. 10) [Section 67A] Punishment for publishing or transmitting of material containing sexually explicit act, etc. in electronic form. 11) [Section 67B] Punishment for publishing or transmitting of material depicting children in sexually explicit act, etc. in electronic form. 12) [Section 72] Breach of confidentiality and privacy.	**10 hours**

Text/ Reference Books

1. Computer Security: Art and Science, M. Bishop, Pearson Education.
2. Information Security: Principles and Practice, M. Stamp, John Wiley & Sons.
3. Cryptography and Network Security, William Stallings, Eastern Economy Edition, PHI.
4. Understanding Cryptography, Paar and Pelzi, Springer.
5. Cryptography and Network Security, Behrouz A Forouzan, McGraw Hill Education.
6. Information Security Principles and Practices by M. Merkow, J. Breithaupt,, Pearson Education.
7. Computer Security: Concepts, Issues and Implementation by A. Basta, W.Halton, Cengage Learning India.

CMS-A-SEC-B-4-2-TH: E-Commerce

Skill Enhancement Course: SEC-B: Choice -2: Theory, Credit:02, Contact hours: 40.

An introduction to Electronic commerce What is E-Commerce (Introduction And Definition), Main activities E-Commerce, Goals of E-Commerce, Technical Components of E-Commerce, Functions of E-Commerce, Advantages and disadvantages of E-Commerce, Scope of E-Commerce, Electronic Commerce Applications, 9 Electronic Commerce and Electronic Business (C2C) (C2G,G2G, B2G, B2P, B2A, P2P, B2A, C2A, B2B, B2C).	**05hours**

The Internet and WWW Evolution of Internet, Domain Names and Internet Organization (.edu, .com, .mil, .gov, .net etc.) , Types of Network, Internet Service Provider, World Wide Web, Internet & Extranet, Role of Internet in B2B Application, building own website, Cost, Time, Reach, Registering a Domain Name, Web promotion, Target email, Banner, Exchange, Shopping Bots.	**10hours**
Internet Security Secure Transaction, Computer Monitoring, Privacy on Internet, Corporate Email privacy, Computer Crime(Laws , Types of Crimes), Threats, Attack on Computer System, Software Packages for privacy, Hacking, Computer Virus(How it spreads, Virus problem, virus protection, Encryption and Decryption, Secret key Cryptography, DES, Public Key Encryption, RSA, Authorization and Authentication, Firewall, Digital Signature(How it Works).	**10hours**
Electronic Data Exchange Introduction, Concepts of EDI and Limitation, Applications of EDI, Disadvantages of EDI, EDI model, Electronic Payment System: Introduction, Types of Electronic Payment System, Payment Types, Value Exchange System, Credit Card System, Electronic Fund Transfer, Paperless bill, Modern Payment Cash, Electronic Cash.	**05hours**
Planning for Electronic Commerce Planning Electronic Commerce initiates, Linking objectives to business strategies, Measuring cost objectives, Comparing benefits to Costs, Strategies for developing electronic commerce web sites.	**05hours**
Internet Marketing: The PROS and CONS of online shopping, The cons of online shopping, Justify an Internet business, Internet marketing techniques, The E-cycle of Internet marketing, Personalization e-commerce.	**05hours**

Text/ Reference Books

1. E-Commerce Concepts, Models, Strategies by G.S.V. Murthy, Himalaya Publishing House.
2. The E-Commerce Book, Teffano Korper and Juanita Ellis, Morgan Kaufmann.
3. E-Commerce 2017, Kenneth C. Laudon and Carol Guercio Traver, Pearson.
4. E- Commerce, Kamlesh K Bajaj and Debjani Nag Tata McGraw-Hill Education.
5. Electronic commerce by Gray P. Schneider , International Student Edition.
6. E-Commerce, Fundamentals and Applications by Henry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang, Wiley Student Edition.

Semester - V				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -11	**Theory**	**CMS-A-CC-5-11-TH**	**Database Management system (DBMS)**	**4**
	Practical	**CMS-A-CC-5-11-P**	**RDBMS lab using My SQL & PHP**	**2**
Core Course -12	**Theory**	**CMS-A-CC-5-12-TH**	**Object Oriented Programming (OOPs)**	**4**
	Practical	**CMS-A-CC-5-12-P**	**OOPs Lab using JAVA**	**2**
Semester -V (DSE) Discipline Specific Elective Course - DSE-A(1&2)& DSE-B(1&2), (Candidates have to opt one course from DSE-A & one course from DSE-B)				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
DSE-A-1	**Theory**	**CMS-A-DSE-A-1-TH**	**Digital Image Processing**	**4**
	Practical	**CMS-A-DSE-A-1-P**	**Image processing Lab**	**2**
DSE-A-2	**Theory**	**CMS-A-DSE-A-2-TH**	**Data Mining & its Application**	**4**
	Practical	**CMS-A-DSE-A-2-P**	**Data Mining Lab**	**2**
DSE-B-1	**Theory**	**CMS-A-DSE-B-1-TH**	**Operation Research (O.R)**	**4**
	Practical	**CMS-A-DSE-B-1-P**	**Operation Research (O.R) Lab**	**2**
DSE-B-1	**Theory**	**CMS-A-DSE-B-2-TH**	**Programming using Python**	**4**
	Practical	**CMS-A-DSE-B-2-P**	**Programming in Python Lab**	**2**

CMS-A-CC-5-11-TH: Database Management System (DBMS).
Core Course- 11: Theory, Credit: 04, Contact hour: 60 hours.

Introduction Drawbacks of Legacy System; Advantages of DBMS; Layered Architecture of Database, Data Independence; Data Models; Schemas and Instances; Database Languages; Database Users, DBA; Data Dictionary.	**04hours**
Entity Relationship(ER) Modeling Entity, Attributes and Relationship, Structural Constraints, Keys, ER Diagram of Some Example Database, Weak and strong Entity Set, Specialization and Generalization, Constraints of Specialization and Generalization, Aggregation.	**04hours**
Relational Model Basic Concepts of Relational Model; Relational Algebra; Tuple Relational Calculus; Domain Relational Calculus.	**08hours**
Integrity Constraints Domain Constraints, Referential Integrity, View.	**04hours**
Relational Database Design Problems of Un-Normalized Database; Functional Dependencies (FD),Derivation Rules, Closure of FD Set, Canonical Cover; Normalization: Decomposition to 1NF, 2NF, 3NF or BCNF Using FD; Lossless Join Decomposition Algorithm; Dependency preservation.	**16hours**
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; Views, Joined Relations; Set Comparisons (All, Some); Derived Relations.	**16hours**

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.	**08hours**

CMS-A-CC-5-11-P: Relational Database Management System
Core Course- 11, Practical, Credit:02, Contact hours: 40 hours.

RDBMS Lab using My SQL & PHP

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.

CMS-A-CC-5-12-TH: Object Oriented Programming System (OOPs)
Core Course- 12: Theory, Credit:04, Contact hours: 60.

Concept of OOPs Difference with procedure oriented programming, Data abstraction and information hiding: Objects, Classes, methods.	**02hours**
Introduction to Java Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods).	**04hours**
Arrays, Strings and I/O Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.	**08hours**
Object-Oriented Programming Overview Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.	**04hours**

Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata. Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.	**14hours**
Exception Handling, Threading, Networking and Database Connectivity Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.	**15hours**
Applets Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.	**13hours**

CMS-A-CC-5-12-P: Object Oriented Programming Lab.
Core Course- 12: Practical, Credit: 02, Contact hours: 40 hours.

OOPs Lab Using JAVA

Text/Reference Books

1. Java: The Complete Reference, Herbert Schildt, McGraw-Hill Education.
2. The Java Language Specification, Java SE by James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley, Published by Addison Wesley.
3. Effective Java by Joshua Bloch, Publisher: Addison-Wesley.
4. Core Java 2 by Cay S. Horstmann, GaryCornell, Volume 1 , Prentice Hall.
5. Programming with Java by E. Balaguruswamy, McGraw Hill.
6. Java: How to Program by Paul Deitel, Harvey Deitel, Prentice Hall.
7. Programming with JAVA by John R. Hubbard, Schaum's Series.

Discipline Specific Elective Course A: DSE-A:
Digital Image Processing/ Data Mining & its Applications.

CMS-A-DSE-A--1-TH: Digital Image Processing.
DSE-A: Choice-1: Theory, Credit:04, Contact hours: 60.

Introduction Image definition and its representation, Pixels, Co-ordinate conventions, Image formats (Study of the image matrix), neighbourhood metrics, Sampling and quantization, Types of distance measure (concept only).	**15hours**
Spatial Domain Image enhancement techniques in spatial domain, Contrast stretching, Histogram Processing, Noise smoothing, Sharpening, Pixel Classification, RGB & Grey image. Transformation: Arithmetic Transformation, Logical Geometric Transformation, Hough Transformation, FFT. Filtering: Spatial domain filters: Convolution, Edge Detection Filters	**15hours**
Thresholding Grey level thresholding, global/ local thresholding, Iterative thresholding, Edge detection operators, Region growing, Split/ merge techniques, Image feature/ primitive extraction, Background correction, Color enhancement.	**15hours**
Image Segmentation Boundary detection based techniques, Point, line detection, Edge detection, Local processing.	**15hours**

CMS-A-DSE-A--1-P: Image Processing Lab.
DSE-A: Choice-1: Practical, Credit:02, Contact hours: 40.

Assignments on Different Image Processing Functions based on Open CV & Python/Scilab

Text/ Reference Books:

1) Digital Image Processing by Gonzalez, Pearson.
2) Digital Image Processing by Jayaraman and Veerakumar, TMH.
3) Digital Image Processing using MATLAB by Gonzalez, Eddins and Woods, McGraw Hill.
4) Digital Image Processing by Annadurai, Pearson.
5) Digital Image Processing; A remote sensing perspective by Jensen, Pearson.
6) Digital Image Processing by Castleman, Pearson.
7) B. Chanda and D. Dutta Majumder, “Digital Image Processing and Analysis”, PHI, New Delhi, 2000.

CMS-A-DSE-A--2-TH: Data Mining and its Applications
DSE-A: Choice-2: Theory, Credit:04, Contact hours: 60.

Introduction Definition of Data Mining, Data pre-processing, Data cleaning, Data transformation, Data Reduction, Data Visualization, Data extraction from large dataset, Data integration, sub-sampling, Feature selection, Scalability issues of data mining algorithms, text mining, web mining.	**15hours**
Classification and Prediction Structural patterns of data, Tools for pattern recognition (preliminary concept), Linear models for classification, Evaluating the accuracy of the classifier or predictor, Bayesian Classification, Training and Test sets, Parametric and Non-parametric Learning, Minimum Distance Classifiers, k-NN rule, Discriminant Analysis, Decision trees. Similarity Measure, Basic hierarchical and non-hierarchical Clustering algorithms, Some Applications, Neural Learning.	**30hours**
Data Warehousing (DWH) Introduction: Definition and description, need for data ware housing, need for strategic information, failures of past decision support systems, Application of DWH.	**15hours**

CMS-A-DSE-A--2-P: Data Mining Lab.
DSE-A: Choice-2: Practical, Credit:02, Contact hours: 40.

Data mining using PYTHON/C

Text/ Reference Books :

1. Data Mining: Practical Machine Learning Tools and Techniques, Ian H. Witten, Eibe Frank, Mark A. Hall, Morgan Kaufman.
2. Pattern Classification and Scene Analysis, R.O. Duba, P.E. Hart and D.G. Stork, Wiley.
3. Pattern Recognition Principles, J.T. Tou and R.C. Gonzalez, Addison-Wesley.
4. Data Mining Concepts and Techniques by Jiawei Han and Micheline Kamber, Morgan Kaufmann Publishers.
5. Data Warehousing, Data Mining and OLAP by Berson, Tata McGraw Hill.
6. Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education.
7. Data mining: Introductory and Advanced Topics by Dunham M H," Pearson Education.
8. Data Mining Concepts, Methods and Algorithms by Mehmed Kantardzic, John Wiley and Sons.

CMS-A-DSE-B--1-TH: Operation Research (O.R.)
DSE-B: Choice-1: Theory, Credit:04, Contact hours: 60.

Introduction Origin and development of operation research, Nature and characteristic features, models in O.R., application of O.R.	**05hours**
Linear Programming Problem Introduction, mathematical formulation of the problem and graphical solution method.	**05hours**
Simplex Method Introduction, computational procedure, artificial variable, problem of degeneracy, application of simplex method.	**20hours**
Duality: Concept, formulation of primal – dual, duality and simplex method, Dual Simplex method.	**10hours**
Transportation Problem: Introduction, mathematical formulation, finding initial basic feasible solution, optimality, degeneracy, unbalanced transportation problem.	**05hours**
Assignment Problem: Introduction, mathematical formulation and solution.	**05hours**
Game Theory: Some basic terminology, Two-person Zero-sum Game, Game without Saddle Point – Mixed strategy, Algebraic method for 2×2 Game	**05hours**
Network Scheduling: Introduction, Critical Path Method (CPM), PERT calculation.	**05hours**

CMS-A-DSE-B-1-P:Operation Research (O.R)Lab using C
DSE-B: Choice-1: Practical, Credit: 02, Contact hours: 40.

Lab sessions related to Simplex Method, Transportation Problem and Assignment Problem.

Text/ Reference Books

1. Operations Research by KantiSwarup, P.K. Gupta, Man Mohan, Sultan Chand & Sons.
2. Schaum's Outline of Operations Research, Richard Bronson and Govindasami Naadimuthu, McGraw-Hill Education.
3. Operations Research: An Introduction, Hamady. A. Taha, TMH.
4. Operations Research: Applications and Algorithms, Wayne L. Winston, Duxbury Press.
5. Operations Research Techniques for Management by V.K.Kapoor, Sultan Chand & Sons.
6. Introduction to Operation Research, Frederick S. Hillier and G. Lieberman, McGraw-Hill Higher Education.

CMS-A-DSE-B--2-TH: Programming using Python 3
DSE-B: Choice-2: Theory, Credit: 04, Contact hour: 60.

Introduction to the Python Interpreted vs. compiled languages. Bytecodes. The importance of whitespace. Variables and the lack of explicit data types and how Python uses the concepts of duck, strong, and static typing, to figure out data types in runtime. The assignment operator, the binding of names to objects, and aliasing. Keywords and their significance.	**04 hours**
Strings: definition, declaration, and immutability, string constants, declaration, and the equivalence of single and double quotes. Multi-line strings. Raw strings. String formatting using the format function and the % operator. f-strings in Python 3.6+. Built-in functions: count, find, replace, upper, lower, strip, etc. Time and space complexities of the functions and operations. **Lists:** definition, declaration, and mutability. Nested lists. Indexing and slicing: same as strings. List comprehensions. The split and join methods. Built-in list functions – append, extend, count, find, index, etc. Time and space complexities of the functions and operations. **Tuples:** definition, declaration, and immutability. Packing and unpacking lists and tuples. The + and * operators on strings, lists, and tuples. Indexing and slicing strings, lists, and tuples.	**06 hours**
Conditionals, Iterators, and Generators **Conditionals:** If, elif, and else statements. Nested conditionals. Containment checking in containers using the in keyword. **Looping constructs:** while and for loops. Flow control using break, continue, and pass. Nested loops. **Generators:** range, zip, sorted, reversed, and enumerate.	**15 hours**
User-defined Functions and Recursion Functions: definition, function signature, positional, default, and keyword arguments. Documentation strings. Unnamed functions – lambda, filter, and map. Recursion: basic idea, implementing recursion, sharing variables across the recursion stack, modifying the size of the recursion stack.	**10 hours**
File Handling and Exception Handling File handling: open and close methods, the different read and write modes. Using the with open approach to files. read, readline, readlines functions. The csv module for efficient read/write of structured data. The pickle module for persistent storage of variables in a program. Exception handling: the popular errors- Name Error, Value Error, Syntax Error, Key Error, Attribute Error, etc, and their cause and effects. Using try-except blocks for graceful handling of exceptions.	**05 hours**
Unordered data types - Sets and Dictionaries **Basic concepts of hashing:** hash functions, open chain, closed chain, advantages and disadvantages compared to conventional ordered data types. The hash() function in Python. **Sets and frozensets:** definition, declaration, mutability, and advantages over lists / tuples.	**05 hours**

Insertion, deletion, union, intersection, and other built-in operations. Time and space complexities of the functions and operations. **Dictionaries:** Concept of keys and values. Immutability requirement for keys. Basic operations on dictionaries. Iterating over the keys and key, value pairs of a dictionary. Dictionary inversions	
Intro to Object Oriented Programming The Python data model, magic methods (__init__, __str__, __eq__, etc) and their utilities, accessing and mutating data, constructors, class methods, and the lack of explicit access modifiers of class methods – naming conventions of private, protected, and public variables and methods. Inheritance: inheriting a parent class, the super() method. Basic multiple inheritance.	**15 hours**

CMS-A-DSE-B--2-P: Python 3 Programming Lab.
DSE-B: Choice-2, Practical, Credit: 02, Contact hours: 40 hours.

Use Python 3.6 or above. Use a text editor sensitive to whitespace like Notepad++, gedit, vim, Sublime Text, and NOT Notepad / WordPad. The following exercises are suggestive in nature.

1. The Interpreter as a calculator. Basic arithmetic operations. Introduction to the simple numeric data types – integers, floating point numbers, Boolean, complex numbers. Inter conversion of data types.
    a. Use the Python prompt as a basic calculator. Explore the order of operations using parentheses.
    b. Explore the various functions in the math module. Eg: find GCD of two numbers, area and perimeter of circle using math.pi, etc.
    c. Exploring the complex data type and their operations, eg: finding the modulus and phase angle of a complex number.
    d. The print function – Printing values. Repeat the previous experiments now using the print function
2. Basic user interactions using the print() and input() functions.
    a. Write a simple python script using the print function in a text editor, save it with the extension ".py". Run it in the terminal / command prompt.
    b. Take input two strings from the user, and print the first one twice, and the other one thrice.
    c. Ask the user to enter two numbers, and output the sum, product, difference, and the GCD.
    d. More programs that test concepts learned in week 1 which involves the usage of the print and input functions.
3. Strings, List, Tuples, the re (regular expression) module
    a. Ask the user for two strings, print a new string where the first string is reversed, and the second string is converted to upper case. Sample strings: "Pets", "party", output: "steP PARTY". Only use string slicing and + operators.
    b. From a list of words, join all the words in the odd and even indices to form two strings. Use list slicing and join methods.
    c. Simulate a stack and a queue using lists. Note that the queue deletion operation won't run in O(1) time.
    d. Explore the 're' module, especially re.split, re.join, re.search and re.match methods.
4. Conditionals, looping constructs, and generators

a. Use list comprehension to find all the odd numbers and numbers divisible by 3 from a list of numbers.
b. Using while loops to do Gaussian addition on a list having an even number of numbers. Print each partial sum. Eg: if the list is [1, 2, 3, 4, 5, 6], the program should output "1 + 6", "2 + 5", and "3+4" in separate lines, and the result of the addition "21". Extend it to handle lists of odd length.
c. Primarily testing using for and while loops.
d. Use (c) to generate a list of primes within a user-given range.
e. Explore the 'key' function of sum( ), min( ), max( ), and sort( ) functions using lambdas.

5. User defined functions
    a. Implement popular sorting algorithms like quick sort and merge sort to sort lists of numbers.
    b. Implement the Pascal's triangle.
    c. Three positive integers a, b, and c are Pythagorean triples if $a^2 + b^2 = c^2$. Write a function to generate all Pythagorean triples in a certain range.
    d. Write two functions that simulate the toss of a fair coin, and the roll of an unbiased 'n' sided die using the random module.
    e. Like (d), but now the coin and the die are not fair, with each outcome having a given probability.
6. File handling, sys, pickle and csv modules
    a. Basic file operations. Explore the different file modes.
    b. Emulate the unix 'cp', 'grep', 'cat' programs in Python. In each case, the user should pass the arguments to the program as command line arguments.
    c. Use pickle for persistent storage of variables
7. Sets and dictionaries
    a. Use sets to de-duplicate a list of numbers, and a string such that they contain only the unique elements
    b. Use the set union and intersection operations to implement the Jaccard and Cosine similarity of two sets.
    c. Use dictionaries to count the word and letter occurrences in a long string of text.
    d. Invert a dictionary such the previous keys become values and values keys. Eg: if the initial and inverted dictionaries are d1 and d2, where d1 = {1: 'a', 2: 'b', 3: 120}, then d2 = {'a': 1, 2: 'b', 120: 3}.
    e. What if the values in (d) are not immutable? Use frozensets. For repeated values, use lists. Eg: if d1 = {1: 'a', 2: 'a', 4: [1, 2]}, then d2 = {'a': [1, 2], frozenset([1, 2]): 4}.
    f. Write a function to generate the Fibonacci numbers in (a) exponential time using the naïve algorithm, and (b) in linear time using dynamic programming (memorization) with a dictionary.
8. Object Oriented Programming
    a. Create a 'Graph' class to store and manipulate graphs. It should have the following functions:
        i. Read an edge list file, where each edge (u, v) appears exactly once in the file as space separated values.
        ii. Add and remove nodes and edges
        iii. Print nodes, and edges in a user readable format
        iv. Computes basic statistics of the graph like degree distribution, clustering coefficient, and the number of connected components.
        v. Finding all the neighbors of a node
        vi. Finding all the connected components and storing them as individual Graph objects inside the class

vii. Finding single source shortest paths using Breadth First Search

b. Make a ‘DiGraph’ class to handle directed graphs which inherits from the ‘Graph’ class. In addition to all of the functionalities of (a), it should support the following operations
   i. Finding the predecessors and successors of a node
   ii. Creating a new ‘DiGraph’ object where all the edges are reversed.
   iii. Finding the strongly connected components
c. Extend (a) and (b) to handle weighted graphs, and implement Dijkstra’s and Floyd-Warshall algorithms to compute the single source and all pairs shortest paths.
d. Use the graph containers in (a), (b), and (c) to implement additional graph algorithms.

Text/ Reference Books

1. Introduction to Computation and Programming Using Python: With Application to Understanding Data, Guttag, John V. MIT Press.
2. Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code, Shaw, Zed A, Addison-Wesley Professional.
3. Think Python 2e. Green Tea Books, Downey, Allen B.
4. Practical Programming: An Introduction to Computer Science Using Python 3.6. Pragmatic Bookshelf, Gries, Paul, Jennifer Campbell, and Jason Montojo.

Semester - VI				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
Core Course -13	**Theory**	**CMS-A-CC-6-13-TH**	**Software Engineering**	**4**
Core Course -14	**Theory**	**CMS-A-CC-6-14-TH**	**Theory of Computation**	**4**
	Practical	**CMS-A-CC-6-14-P**	**Project Work**	**4**
Semester - VI (DSE) **Discipline Specific Elective Course - DSE-A(3&4)& DSE-B (3&4)** **(Candidates have to opt one course from DSE-A & one course from DSE-B)**				
Course	**Type**	**Course Code**	**Course Name**	**Credit**
DSE-A-3	**Theory**	**CMS-A-DSE-A-3-TH**	**Embedded Systems**	**4**
	Practical	**CMS-A-DSE-A-3-P**	**Embedded Systems Lab**	**2**
DSE-A-4	**Theory**	**CMS-A-DSE-A-4-TH**	**Multimedia and its Application**	**4**
	Practical	**CMS-A-DSE-A-4-P**	**Multimedia and its Application Lab**	**2**
DSE-B-3	**Theory**	**CMS-A-DSE-B-3-TH**	**Introduction to Computational Intelligence**	**4**
	Practical	**CMS-A-DSE-B-3-P**	**Computational Intelligence Lab**	**2**
DSE-B-4	**Theory**	**CMS-A-DSE-B-4-TH**	**Advance Java**	**4**
	Practical	**CMS-A-DSE-B-4-P**	**Advance Java Lab**	**2**

CMS-A-CC-6-13-TH: Software Engineering.
Core Course-13: Theory, Credit:04, Contact hours 60.

Introduction Defining system, open and closed system, modeling of system through computer hardware, communication systems, external agents and software systems; Importance of Engineering Methodology towards computerization of a system.	**03 hours**
Software Life Cycle Classical and Iterative Waterfall Model; Spiral Model; Prototype Model; Evolutionary model and its importance towards application for different system representations, Comparative Studies.	**07 hours**
Software Requirement and Specification Analysis Requirements Principles and its analysis principles; Specification Principles and its representations Software Design Analysis – Different level of DFD Design, Physical and Logical DFD, Use and Conversions between them, Decision Tables and Trees, Structured analysis, Coupling and Cohesion of different modules Software Cost Estimation Modeling –COCOMO.	**23 hours**
Software Testing Software Verification and Validation; Testing objectives, Testing Principles, Testability; Error and Faults; Unit Testing, White Box and Blank Box Testing, Test Case Design: Test Vector, Test Stub.	**17 hours**
Software Quality Assurances Concepts of Quality, Quality Control, Quality Assurance, IEEE Standard for Statistical Software Quality Assurances (SSQA) criterions.	**10 hours**

Text/ Reference Books

1. Software Engineering: A Practitioner's Approach by R.S. Pressman, McGraw-Hill.
2. An Integrated Approach to Software Engineering by P. Jalote, Narosa Publishing House.
3. Software Engineering by K.K. Aggarwal and Y. Singh, New Age International Publishers.
4. Software Engineering by I. Sommerville, Addison Wesle.
5. Software Engineering for Students by D. Bell, Addison-Wesley.
6. Fundamentals of Software Engineering by R. Mall, PHI.

CMS-A-CC-6-14-TH: Theory of Computation.
Core Course-14: Theory, Credit:04, Contact hours: 60.

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, Finite and Infinite state machines, Finite Automaton, Deterministic and Non-Deterministic Finite automaton, Non-Deterministic to equivalent Deterministic Automaton-Optimized and Non-optimized technique ideas and algorithms, Acceptability of String by a Finite Automaton.	**15 hours**
Formal Languages and Grammar Introduction to Formal Grammar and Language, 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, Ambiguous Grammar and Language, Designing of Grammar for a language, Find the Language for given Grammar, Definition and basic idea about Push Down Automaton.	**15 hours**
Regular Expression: Basic Idea and Definition, Regular Expression basic Identities, Arden's Theorem – Statement (without Proof) and application for reduction of equivalent regular expressions, 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.	**15 hours**
Turing Machine Concepts of Turing Machine, Formal Definitions, Classifications – Deterministic and Non-Deterministic Turing Machines, Simple Design of Turing Machines: Odd / even count and concepts of Universal Turing Machines, Difference and Similarities between Turing Machine and a General Purpose Computer, Definition and significant of Halting Problem in Turing Machine.	**15 hours**

Text/ Reference Books:

1. Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, 3rd Edition, Pearson.
2. Theory of Computer Science (Automata, Languages & Computation) by K L P Misra&

N Chandrasekharan, 3rd Edition, PHI.
3. Introduction to Theory of Computation by MichealSipser, 3rd Edition, Cengage Learning.
4. Switching and Finite Automata Theory by ZviKohavi, Niraj.K.Jha, 3rd Edition, TMH.
5. Formal Language and Automata, P. Linz, Narosa

CMS-A-CC-6-14-P: Project Work
Core Course-14, Practical, Credit:04, Contact hours: 60.

Candidates have to do their project in any relevant topic, under the supervision

of teachers.

Discipline Specific Elective Course A: DSE-A

CMS-A-DSE-A--3-TH: Embedded Systems
DSE-A: Choice-3: Theory, Credit:04, Contact hours: 60.

Introduction to 8051 Overview of Microcontroller, Memory, I/O interface Intel Microcontroller 8051: Architecture, Peripheral Interface Controller (PIC).	**15 hours**
Assembly Language Programming Instruction set, Addressing Modes, Jump, Loop and Call instructions, I/O Manipulation, Serial communication, Arithmetic and logical instructions.	**10 hours**
Introduction to Embedded System Programming Data types and time delays, I/O programming, Logic operations, Data conversions, Data serialization, Interrupt programming, LCD and Keyboard interfacing, ADC, DAC, sensors interfacing, interfacing 8255, I/O interfacing for 8051, interfacing 8255, 8257, 8259/ 8279, ADC, DAC, Motor control using 8051 C.	**15 hours**
Programmable logic devices and Hardware description Language PAL, PLA, PLD, ASIC, FPGA (Qualitative study).	**10 hours**
Hardware Description Language (VHDL): Basic Terminology, Entity Declaration, Architecture body, Configuration and package declaration, Package body, Model analysis and Simulation. Basic Language elements, Behavioral Model, Dataflow Model, Structural Model, Subprogram and overloading, Applications.	**15 hours**

CMS-A-DSE-A--3-P: Embedded Systems Lab.
DSE-A: Choice-3, Practical, Credit: 02, Contact hours: 40 hours

Practical: Sample practical problems can be included related to theory.
1. Assembly Language Programming related to Microcontroller 8051.
2. Programming Using Embedded C for 8051.
3. VHDL programs for construction and simulation of various digital circuits.

Text/ Reference Books:

1. An Embedded software primer,David E.Simon, Pearson Education.
2. The 8051 Microcontroller, Kenneth J. Ayala, Thomson.
3. Embedded Systems, Raj Kamal, TMH.
4. Microcontroller, Raj Kamal, Pearson Education.
5. A VHDL Primer, J. Bhasker, Prentice Hall
6. FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version, Pong P. Chu, Wiley-Interscience.

CMS-A-DSE-A--4-TH: Multimedia and its Applications
DSE-A: Choice-4, Theory, Credit:04, Contact hours: 60.

Multimedia Introduction to multimedia, Components, uses of multimedia.	**04 hours**
Making Multimedia Stages of a multimedia project, requirements to make good multimedia, Multimedia Hardware - Macintosh and Windows production Platforms, Hardware peripherals - Connections, Memory and storage devices, Multimedia software and Authoring tools.	**06 hours**
Text Fonts & Faces, Using Text in Multimedia, Font Editing & Design Tools, Hypermedia & Hypertext.	**04 hours**
Images Still Images – Bitmaps, Vector Drawing, 3D Drawing & rendering, Natural Light & Colors, Computerized Colors, Color Palettes, Image File Formats.	**06 hours**
Sound Digital Audio, MIDI Audio, MIDI vs Digital Audio, Audio File Formats.	**06 hours**
Video How Video Works, Analog Video, Digital Video, Video File Formats, Video Shooting and Editing.	**06 hours**
Animation Principle of Animations. Animation Techniques, Animation File Formats.	**08 hours**
Multimedia System An overview of multimedia system and media streams, Source representation and compression techniques text, speech and audio, still image and video, Graphics and animation.	**10 hours**
Multi-modal Communication Video conferencing, networking support, Trans-coding.	**10 hours**

CMS-A-DSE-A--4-P: Multimedia and its Applications Lab.
DSE-A: Choice-4: Practical, Credit:02, Contact hour: 40.

Sample practical problems can be included related to theory.

Text/ Reference Books:

1. Multimedia: Making it work by Tay Vaughan, TMH.
2. Multimedia: Computing, Communications Applications by R Steinmetz and K Naharstedt, Pearson.
3. Multimedia Handbook by Keyes, TMH.
4. Multimedia System Design by K. Andleigh and K. Thakkar, PHI.

Discipline Specific Elective Course B: DSE-B.
Introduction to Computational Intelligence/ Advanced Java.

CMS-A-DSE-B--3-TH:Introduction to Computational Intelligence
DSE-B: Choice-3, Theory, Credit:04, Contact hours: 60.

Introduction Introduction to Artificial Intelligence, Brief History and Application, Structures and Strategies for state space search- Data driven and goal driven search, Heuristic search, Depth First and Breadth First search, Iterative deepening, A* algorithm, Game playing (Minimax), Rule-based system, Semantic Nets, Frames, Scripts, Conceptual Dependency, Introduction to PROLOG.	**20 hours**
Neural Network Basics of Artificial Neural Network, Characteristics and Comparison with biological neural network, Basic model of Artificial Neural Network: Single layer Perceptron model, Learning, Feed Forward Neural Network, Error, Back Propagation and weight updation, Perceptron, Bayesian Networks, Neural computational model- Hopfield Nets. .	**20 hours**
Rough sets Basic difference between Rough sets and Fuzzy sets	**02 hours**
Fuzzy Logic and Application Fuzzy sets, application – basic operations, Properties, Fuzzy Relations, Fuzzy inference, Notion of Fuzziness, Operations on Fuzzy sets, Fuzzy Numbers, Brief overview of crisp sets, Crisp relations, Fuzzy relations, Max*-composition of fuzzy relation, Max*-transitive closure, Probability measures of fuzzy events, Fuzzy expected value, Approximate reasoning, Different methods of role aggregation and defuzzification.	**18 hours**

CMS-A-DSE-B-3-P:Computational Intelligence Laboratory
DSE-B: Choice 3, Practical, Credit: 02, Contact hours: 40.

Computational intelligence lab using Prolog / LISP

Text/ Reference Books:

1. Pattern Recognition and Machine Learning, Christopher M. Bishop.
2. Artificial Intelligence, E, Rich and K. Knight, Tata McGraw Hill.
3. A Brief Introduction to Neural Network, David Kriesel.
4. Fuzzy Set Theory – and its Applications, H.J. Zimmermann.
5. Rough Set Data Analysis : A road to Non-invasive Knowledge Discovery, Methods, Ivo Duntsch & Gunther Gediga.

6. An Introduction to Neural Computing: Theory and Practice, P.D. Wassermann, Van Nostrand Reinhold, New York, 1989.
7. Artificial Neural Networks, B. Yegnarayana, Prentice Hall of India.

CMS-A-DSE-B--4-TH: Advanced Java
DSE-B: Choice-4, Theory, Credit:04, Contact hours: 60.

Basics of Servlet Servlet: What and Why? Servlet API, Servlet interface, Generic Servlet, Http Servlet, Servlet life cycle, Servlet request methods, Servlet collaboration, Servlet config.	**10 hours**
Session Management What is a session? Why is it required? Creating a session? Session information passing mechanisms between client and server - Cookies, Rewriting; Destroying a session. .	**04 hours**
Basics of JSP Life cycle of JSP; JSP API;JSP tags, directives, scripting elements, implicit objects, exception handling, action elements; MVC.	**10 hours**
Design Pattern Singleton; DAO; DTO; MVC; Front controller; Factory method; Collection framework.	**10 hours**
Javascript Introduction to Javascript; Ways to use Javascript; Working with events; Client-side validation.	**10 hours**
JQuery Introduction to JQuery; Validation using JQuery; JQuery forms; JQuery examples; Key services of the application server.	**06 hours**
Spring Framework Spring Core (Basic Concepts); Spring AOP; Spring JDBC; Spring MVC; Spring Boot and Spring Data; Spring ORM.	**10 hours**

Text/ Reference Books:

1. Object-Oriented Software Development Using Java. Xiaoping Jia. Addison Wesley,
2. Head First Object-Oriented Analysis and Design. Brett D. McLaughlin, Gary Pollice, and Dave West. O'Reilly.
3. Head First Design Patterns. Eric Freeman and Elizabeth Freeman. O'Reilly
4. Head First Servlets & JSP, O'Reilly.
5. Murach's Java Servlets and JSP, Murach.
6. Core Servlets and Javaserver Pages: Core Technologies, Marty Hall and Larry Brown, Prentice Hall.
7. Enterprise JavaBeans 3.0, Richard Monson-Haefel and Bill Burke, O'Reilly.
8. Java Design Pattern Essentials, Tony Bevis, Ability First Limited
9. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley Professional

10. Getting started with Spring Framework – Ashish Sarin, J Sharma, Createspace
11. Spring in Action - Craig Walls, Manning Publications

CMS-A-DSE-B-4-P: Advanced Java Laboratory
DSE-B: Choice 4, Practical, Credit:02, Contact hours: 40.

Advanced Java Laboratory based on the following:

(i) Write programs in Java using Servlets:
   a. To invoke servlets from HTML forms.
   b. To invoke servlets from Applet Programs using cookies.
(ii) Programs with session tracking.
(iii) Create dynamic web pages, using Servlets and JSP.
(iv) Programs using JDBC with create, insert table data.
(v) Implementing MVC with Request Dispatcher.
(vi) Writing a web service.

Text/ Reference Books

1. Core Servlets and Javaserver Pages: Core Technologies, Marty Hall and Larry Brown, Prentice Hall.
2. JavaScript: The Definitive Guide, David Flanagan, O'Reilly.
3. Enterprise JavaBeans 3.0, Richard Monson-Haefel and Bill Burke, O'Reilly.
4. JavaScript and JQuery: Interactive Front-End Web Development, Jon Duckett, Wiley.
5. Professional JavaScript for Web Developers, Nicholas C. Zakas, Wrox.
6. Java Design Pattern Essentials, Tony Bevis, Ability First Limited.
7. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley Professional.

UNIVERSITY
OF
CALCUTTA

SYLLABUS
of
Bachelor of Science (General)
in
Computer Science (CMSG)
Choice Base Credit System (CBCS)
2018

Semester-wise courses for B.Sc. (General)

	Sem-1	Sem-2	Sem-3	Sem-4	Sem-5	Sem-6
Core Course (CC)	**CC-1**	**CC-2**	**CC-3**	**CC-4**		
AECC	**AECC-1**	**AECC-2**				
Skill Enhancement course (SEC)			**SEC-A**	**SEC-B**	**SEC-A**	**SEC-B**
Total No. of Courses & marks	**4x100 =400**	**4x100 =400**	**4x100 =400**	**4x100 =400**	**4x100=400**	**4x100=400**
Total Credits	**20**	**20**	**20**	**20**	**20**	**20**

Computer Science General (CMSG) Syllabus

Courses	Topics	Credit
CMS-G-CC-1-1-TH Sem-1-Core Course-1 Theory	Computer Fundamentals and Digital Logic Design	04
CMS-G-CC-1-1-P Sem-1-Core Course-1 Practical	Word Processing, Spreadsheet, Presentation and Web design by HTML	02
CMS-G-CC-2-2-TH Sem-2- Core Course-2 Theory	Algorithm and Data Structure	04
CMS-G-CC-2-2-P Sem-2-Core Course-2 Practical	Programming with C	02
CMS-G-CC-3-3-TH Sem-3- Core Course-3 Theory	Computer Organization	04
CMS-G-CC-3-3-P Sem-3-Core Course-3 Practical	Programming using PYTHON	02
CMS-G-CC-4-4-TH Sem-4- Core Course-4 Theory	Operating Systems	04
CMS-G-CC-4-4-P Sem-4-Core Course-4 Practical	Shell Programming (Linux)	02
Skill Enhancement Courses (SEC-A & B): Any one topic to be opted from SECA either in Semester-3 or in Semester-5. Any one topic to be opted from SECB either in Semester-4 or in Semester-6.		
CMS-G-SEC-A-X-1-TH	Communication, Computer Network and Internet	02
CMS-G-SEC-A-X-2-TH	Software Engineering	02
CMS-G-SEC-B-X-1-TH	Multimedia and its Applications	02
CMS-G-SEC-B-X-2-TH	Information Security	02
Discipline Specific Elective- A (DSE- A): Candidate has to opt any 2 of the following topics		
CMS-G-DSE-A-5-1-TH	Data base Management System (DBMS)	04
CMS-G-DSE-A-5-1-P	DBMS Lab using SQL	02
CMS-G-DSE-A-5-2-TH	Operation Research	04
CMS-G-DSE-A-5-2-P	Operation Research Lab using C	02
CMS-G-DSE-A-5-3-TH	Computer Graphics	04
CMS-G-DSE-A-5-3-P	Computer Graphics Lab using C	02
Discipline Specific Elective- B (DSE- B): Candidate has to opt any 2 of the following topics		
CMS-G-DSE-B-6-1-TH	Embedded Systems	04
CMS-G-DSE-B-6-1-P	Embedded Systems Lab.	02
CMS-G-DSE-A-6-2-TH	Object Oriented Programming	04
CMS-G-DSE-A-6-2-P	Object Oriented Programming by Java	02
CMS-G-DSE-A-6-3-TH	Computational Mathematics	04
CMS-G-DSE-A-6-3-P	Computational Mathematics Lab using C	02

Semester –I

Courses	Topics	Periods	Credit
CMS-G-CC-1-1-TH Sem-1-Core Course-1 Theory	Computer Fundamentals and Digital Logic Design	60 hours	04
CMS-G-CC-1-P Sem-1-Core Course-1 Practical	Word Processing, Spreadsheet, Presentation and Web design by HTML	40 hours	02

CMS-G-CC-1-1-TH: Computer Fundamentals and Digital Logic Design
Core Course- 1: Theory: 60 Hours

Group A: Computer Fundamentals **(**20 hours)

General Concepts:

Introduction to Computer and Problem Solving: Information and Data

Hardware: CPU, Primary and Secondary storage, Cache Memory, I/O devices, Bus structure, BIOS

Software: Systems and Application.

Generation of Computers: Super, Mainframe, Mini and Personal Computer, Work stations, Parallel machines (concept only).

Introduction to Programming Languages: Machine Language, Assembly Language, High Level Language.

Problem Solving: Flow Charts, Decision Tables and Pseudo codes.

System Software: Classifications- Operating Systems (OS); Translators – Compilers and Interpreters, Preprocessors, Assemblers, Loaders, Linkers, Line and Screen Editors, other utilities.

Virus: Concept, Detection and Protection

Group B: Digital Logic Design (40 hours)

Number Systems and Codes: (08 hours)

Number representation: Weighted Codes, Non-weighted codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Conversion of bases. Complement notions: 1's complement, 2's complement, Binary Arithmetic, Binary Codes: Gray, Alphanumeric, ASCII, EBCDIC; Single Error-Detecting and Correcting Codes, Hamming Codes, Fixed point, Floating point representation.

Boolean Algebra: (08 hours)

Fundamentals of Boolean Algebra, Switches and Inverters, Functionally Complete Gates (AND, OR, NOT), NAND, NOR, Boolean Function. De Morgan's Theorem, Min-term, Max term, Truth tables and minimization of Logic expression up to four variables, Boolean Algebraic and K-map methods of Logic circuit synthesis, two-level and multi-level.

Digital Electronics: (24 hours)

Combinational Circuits: Realization of AND and OR Gates using diodes and NOT Gate using transistors, Half adder and Full Adder (3 & 4 bit), Multi-bit adders – Ripple carry and Carry Look Ahead Adder, Adder/subtractor, BCD-Adder, Data selectors/multiplexers – expansions, reductions, function realization, universal function realization, multi-function realization,

Decoders: function realization, De-multiplexer and function realization, Encoder, Priority Encoder, Parity bit Generator/checker, Gray Code Generator, Code Converters, Keyboard encoder, Seven segment display unit, Comparators.

Sequential Circuits: Model of Sequential computing, Difference between Combinational and Sequential circuit, RS-Latch: using NAND and NOR Gates, Digital Clock – Duty Cycle, Rising time, Falling time, Clocked Flip Flops - SR, JK, D, T, Level Trigger and Edge Trigger, Excitation Functions of each flip-flops, Flip-flops with Preset and Clear, Application of Flip-flops: Asynchronous Counter (UP/DOWN) up to 4 bit counter, Decade Counter, Mod – n Counter, Finite State machine Model – State Transition Diagram and Table, Synchronous Counters – different mod-n counters, Ring counter, Registers: Registers with parallel load, Shift Registers.

CMS-G-CC-1-1-P: Word Processing, Spreadsheet, Presentation and Web design by HTML
Core Course- 1: Practical: 40 Hours

Word Processing: (05 hours)
Document creation, saving, editing; Formatting text and paragraphs**;** header and footers**;** clipart, tables; tools, Inserting images, files; mail merge; margins; Hyphenation; page setups; OLE; index and references; comments; templates; macros.

Spreadsheet: (05 hours)
Workbook, worksheets, cell; address; entering, editing, formatting, filtering, sorting worksheet data; printing; charts; functions and formula; macros; importing, exporting files.

Presentation: (05 hours)
Slides; formatting; wizard, layout; word art; animation.

Web Design: (25 hours)
Web page design can be taught in the laboratory classes by using HTML.
Basic Tags and Document structure, HTML Tags, Head Tags, Title Tags, Introduction to HTML and Web design, How to create simple Web page, How to format text, Create Table, Adding Web link and Images, Forms, Adding styles and classes to web pages, Borders and Background, Adding Video and Graphics.

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. Computer Fundamentals, Anita Goel, Pearson Education.
7. Introduction to Computer Science, P.K.Sinha, P.Sinha, BPB Publication.

Semester –II

Courses	Topics	Periods	Credit
CMS-G-CC-2-2-TH Sem-2-Core Course-2 Theory	Algorithms and Data Structure	60 hours	04
CMS-G-CC-2-2-P Sem-2-Core Course-2 Practical	Programming with C	40 hours	02

CMS-G-CC-2-2-TH: Algorithms & Data Structure
Core Course- 2: Theory: 60 hours

Introduction: Algorithms, ADT. (04 hours)

Arrays: (8 hours)
One dimensional and Two Dimensional Arrays, Row Major and Column Major Forms.

Linked List: (10 hours)
Singly, Circular and Doubly Linked List; Operations Like Insertion, Deletion, Searching.

Stacks and Queues: (14 hours)
Concepts of Stack and Queue; Insertion and Deletion of Elements; Array and Linked Representation: Prefix, Infix and Postfix Notation; Postfix and Prefix Expression Evaluation using stack, Infix to Postfix conversion using stack.

Searching: (04 hours)
Algorithm of Sequential, Binary Search Techniques.

Sorting: (10 hours)
Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort

Tree: (10 hours)
Binary tree; Pre-order, In-order and Post-order traversal; Binary Search Tree (BST): Creation, Insertion and Deletion

CMS-G-CC-2-2-P: Programming with C
Core Course- 2: Practical: 40 hours

Basic Structure: Character set, keywords, identifiers, constants, variables and type declaration. Sample programs, preprocessor.

Operators: Arithmetic, Relational, Logical, Assignment, Increment and Decrement, Conditional, comma; operator precedence and associatively; arithmetic expression-evaluation and type conversion. Character I/O, Escape sequence and formatted I/O.
Branching and Looping: if, if-else, while, do-while, for.

Arrays: One-dimensional and Two-dimensional, Different types of uses. String handling with arrays – read and write, concatenation, comparison, string functions.

User defined functions: Need; Call by Reference and Call by value; return values and types; nesting of functions; recursion.

Structures: Initialization; arrays of a structure, arrays within structures, structure within structure.

Pointers: Declaration and initialization; operators; pointer arithmetics; accessing variables, pointer & arrays, strings, functions.

File handling: Opening & Closing, I/O.

Examples:

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. 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.
25. Add two distances in meter kilometer system using structure.
26. Add two complex numbers using structures.
27. Calculate the difference between two time periods using structures.

These are only examples; more can be included related to the theory. Use open source C compiler.

Text/ Reference Books:

1. Data Structure , Liptsuitz, S. Outline Series.
2. Data Structure, Ellis Horowitz and Sartaz Sahani, Galgotia.
3. Data Structure using C, S.K.Bandyopadhyay and K.N.Dey, Pearson Education.
4. Data Structure and Algorithm Analysis in C, Mark Allen Weiss, Pearson Education.
5. Programming in C. E. Balagurusamy, TMH.
6. Let us C, Y. Kanetkar, BPB Publication.

Semester –III

Courses	Topics	Periods	Credit
CMS-G-CC-3-3-TH Sem-3-Core Course-3 Theory	Computer Organization	60 hours	04
CMS-G-CC-3-3-P Sem-3-Core Course-3 Practical	Programming using Python	40 hours	02

CMS-G-CC-3-3-TH: Computer Organization
Core Course- 3: Theory: 60 hours

Basic Computer Organization: (15 hours)
IAS Computer, Von Neumann Computer, System Bus. Instruction Cycle, Data Representation, Machine cycle, CPU Organization: Arithmetic and Logic Unit, Control Unit, CPU Registers, Instruction Registers, Program Counter, Stack Pointer, CISC & RISC processors.

Instruction: (02 hours)
Operation Code and Operand, One, Two and Three address instruction. Instruction types.

Control Unit: (05 hours)
Control Structure, Hardwired Control and Micro programmed Control: Basic Concept, Parallelism in Micro-instruction.

ALU: (10 hours)
Basic Structure of ALU, Addressing mode, Instruction Formats, Handling of interrupts and subroutines, Combinational ALU, 2's Complement Addition, Subtraction Unit, Booth's Algorithm for multiplication and division.

Memory: (15 hours)
Types of Memory: Primary and Secondary; RAM, ROM, EPROM, EEPROM, DRAM, SRAM, PLA. Different storage technology; Memory Hierarchy: CPU Register, Cache Memory, and Virtual Memory.

I/O: (08 hours)
Polling, Interrupts, DMA, I/O Bus and Protocol, Memory mapped I/O and I/O mapped I/O, I/O system organization and interfacing, Bus: SCSI, PCI, USB, Bus arbitration.

Computer Peripherals: (05 hours)
VDU, Keyboard, Mouse, Printer, Scanner etc.

Text/ Reference Books:
1. Computer Architecture and Organizations, J.P.Hayes, TMH.
2. Computer System Architecture, M. Morris Mano, PHI.
3. Computer Organization and Architecture, William Stallings, Pearson Education.
4. Computer Architecture and Logic Design, Thomas C. Bartee, Mc. Graw Hill.

CMS-G-CC-3-3-P: Programming using Python
Core Course- 3: Practical: 40 hours

Open Source Computer Programming Language Python 3

Introduction to the Python: (2 hours)
Interpreted v. compiled languages. The importance of whitespace.Variables and the assignment operator, the binding of names to objects, and aliasing. Keywords and their significance.

Ordered Datatypes - Strings, Lists and Tuples: **(**6 hours)
Strings: definition, declaration, and immutability, string constants, declaration, and the equivalence of single and double quotes. Multi-line strings. Raw strings. String formatting using the format function and the % operator. f-strings in Python 3.6+. Built-in functions: count, find, replace, upper, lower, strip, etc. Time and space complexities of the functions and operations.

Lists: definition, declaration, and mutability. Nested lists. Indexing and slicing: same as strings. List comprehensions. The split and join methods. Built-in list functions – append, extend, count, find, index, etc. Time and space complexities of the functions and operations.

Tuples**:** definition, declaration, and immutability. Packing and unpacking lists and tuples.

The + and * operators on strings, lists, and tuples. Indexing and slicing strings, lists, and tuples.

Conditionals and Iterators: (12 hours)
Conditionals: If, elif, and else statements. Nested conditionals. Containment checking in containers using the in keyword.

Looping constructs: while and for loops. Flow control using break, continue, and pass. Nested loops.

User-defined Functions and Recursion (10 hours)
Functions: definition, function signature, positional, default, and keyword arguments. Documentation strings.

Recursion: basic idea, implementing recursion, sharing variables across the recursion stack, modifying the size of the recursion stack.

File Handling and Exception Handling (5 hours)
File handling: open and close methods, the different read and write modes. Using the with open approach to files. read, readline, readlines functions.

Exception handling: the popular errors- NameError, ValueError, SyntaxError, KeyError, AttributeError, etc, and their cause and effects. Using try-except blocks for graceful handling of exceptions.

Unordered data types - Sets and Dictionaries (5 hours)
Basic concepts of hashing: hash functions, open chain, closed chain, advantages and disadvantages compared to conventional ordered data types. The hash() function in Python.

Sets and frozensets: definition, declaration, mutability, and advantages over lists / tuples. Insertion, deletion, union, intersection, and other built-in operations. Time and space complexities of the functions and operations.

Dictionaries: Concept of keys and values. Immutability requirement for keys. Basic operations on dictionaries. Iterating over the keys and key, value pairs of a dictionary. Dictionary inversions.

Suggested lab exercises

Use Python 3.6 or above. Use a text editor sensitive to whitespace like Notepad++, gedit, vim, Sublime Text, and NOT Notepad / WordPad.

1. The Interpreter as a calculator. Basic arithmetic operations. Introduction to the simple numeric data types – integers, floating point numbers, Boolean, complex numbers. Interconversion of datatypes.
    a. Use the Python prompt as a basic calculator. Explore the order of operations using parentheses.
    b. Explore the various functions in the math module. Eg: find GCD of two numbers, area and perimeter of circle using math.pi, etc.
    c. Exploring the complex data type and their operations, eg: finding the modulus and phase angle of a complex number.
    d. The print function – Printing values. Repeat the previous experiments now using the print function
2. Basic user interactions using the print() and input() functions.
    a. Write a simple python script using the print function in a text editor, save it with the extension ".py". Run it in the terminal / command prompt.
    b. Take input two strings from the user, and print the first one twice, and the other one thrice.
    c. Ask the user to enter two numbers, and output the sum, product, difference, and the GCD.
    d. More programs that test concepts learned in week 1 which involves the usage of the print and input functions.
3. Strings, List, Tuples, the re (regular expression) module
    a. Ask the user for two strings, print a new string where the first string is reversed, and the second string is converted to upper case. Sample strings: "Pets", "party", output: "steP PARTY". Only use string slicing and + operators.
    b. From a list of words, join all the words in the odd and even indices to form two strings. Use list slicing and join methods.
    c. Simulate a stack and a queue using lists. Note that the queue deletion operation won't run in O(1) time.
4. Conditionals, looping constructs, and generators
    a. Use list comprehension to find all the odd numbers and numbers divisible by 3 from a list of numbers.
    b. Using while loops to do Gaussian addition on a list having an even number of numbers. Print each partial sum. Eg: if the list is [1, 2, 3, 4, 5, 6], the program should output "1 + 6", "2 + 5", and "3+4" in separate lines, and the result of the addition "21". Extend it to handle lists of odd length.
    c. Primality testing using for and while loops.
    d. Use (c) to generate a list of primes within a user-given range.
5. User defined functions

a. Implement popular sorting algorithms like quicksort and merge sort to sort lists of numbers.
b. Implement the Pascal's triangle.
c. Three positive integers a, b, and c are Pythagorean triples if $a^2 + b^2 = c^2$. Write a function to generate all Pythagorean triples in a certain range.
d. Write two functions that simulates the toss of a fair coin, and the roll of an unbiased 'n' sided die using the random module.
e. Like (d), but now the coin and the die are not fair, with each outcome having a given probability.

6. File handling, sys, pickle and csv modules
    a. Basic file operations. Explore the different file modes.
    b. Emulate the unix 'cp', 'grep', 'cat' programs in Python. In each case, the user should pass the arguments to the program as command line arguments.
7. Sets and dictionaries
    a. Use sets to de-duplicate a list of numbers, and a string such that they contain only the unique elements
    b. Use the set union and intersection operations to implement the Jaccard and Cosine similarity of two sets.
    c. Use dictionaries to count the word and letter occurrences in a long string of text.
    d. Invert a dictionary such the previous keys become values and values keys. Eg: if the initial and inverted dictionaries are d1 and d2, where d1 = {1: 'a', 2: 'b', 3: 120}, then d2 = {'a': 1, 2: 'b', 120: 3}.
    e. What if the values in (d) are not immutable? Use frozensets. For repeated values, use lists. Eg: if d1 = {1: 'a', 2: 'a', 4: [1, 2]}, then d2 = {'a': [1, 2], frozenset([1, 2]): 4}.
    f. Write a function to generate the Fibonacci numbers in (a) exponential time using the naïve algorithm, and (b) in linear time using dynamic programming (memoization) with a dictionary.

References

1. Guttag, John V. *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. MIT Press, 2016. (2nd edition)

2. Shaw, Zed A. *Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code*. Addison-Wesley Professional, 2017.

3. Downey, Allen B. *Think Python 2e*. Green Tea Books, 2015. (2nd edition)

4. Gries, Paul, Jennifer Campbell, and Jason Montojo. *Practical Programming: An Introduction to Computer Science Using Python 3.6*. Pragmatic Bookshelf, 2017.

Semester –IV

Courses	Topics	Periods	Credit
CMS-G-CC-4-4-TH Sem-4-Core Course-4 Theory	Operating Systems	60 hours	04
CMS-G-CC-4-4-P Sem-4-Core Course-4 Practical	Shell Programming (Unix/ Linux)	40 hours	02

CMS-G-CC-4-4-TH: Operating Systems
Core Course- 4: Theory: 60 hours

System Software: (04 hours)
Introduction: Different System Softwares

Introduction (08 hours)
Basic OS functions, types of operating systems- batch processing, multiprogramming, time sharing, multiprocessing, distributed and real time systems.

Operating System Organization (02 hours)
Processor and user modes, kernels, system calls and system programs.

Process (18 hours)
System view of the process and resources, process control block, I/O and CPU bound process, process hierarchy, concept of threads, Process Scheduling: Preemptive and non-preemptive scheduling, Long term scheduling, short term/CPU scheduling (FCFS, SJF, SRJF, RR and priority) and medium term scheduling
Process Synchronization: Concurrent processes, critical section, semaphores and application, methods for inter-process communication;

Deadlock: (09 hours)
Definition, Prevention, Avoidance, Detection, Recovery.

Memory Management (14 hours)
Physical and logical address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory

File and I/O Management (05 hours)
Directory structure, file operations, file allocation methods, disk management.

CMS-G-CC-4-4-P: Shell Programming (Linux)
Core Course- 4: Practical: 40 hours
Examples:

1. Write a shell script to convert the content of a file from lower case to upper case.
2. Write a shell script to count the words, lines and characters of a given file. File name should be provided at run time.
3. Write a shell script that take a word from user and find out the frequency of the word in a given file.
4. Write a shell script that gets executed at the moment of user login and it displays Good Morning, Good afternoon, Good Evening, Good Night, depending upon the time at which the user logs on.

5. Write a shell script to print Pascal diamond.
6. Write a shell script to find a number using sequential search method.
7. Write a shell script to find a number using binary search technique.
8. Write a shell script to sort a set of integer numbers using bubble sort.
9. Write a shell script to find out the factorial of a given number.
10. Write a shell script to reverse a string and check whether it is a palindrome.
11. Write a shell script to find the roots of a quadratic equation $ax^2 + bx + c = 0$, considering all possible cases.
12. Write a shell script for menu based system to insert records for employees with employee ID, name, designation, salary in a data file, also display records when necessary. Display salary for the employee asked.

These are only examples, more can be included.

Text/Reference Books:

1. Operating Systems, H.M.Deitel, Pearson Education.
2. Operating System Concepts, A.Silberschatz, Peter B. Galvin, G.Gagne, John Wiley and sons.
3. Unix Shell Programming, Y. Kanetkar.
4. Your Unix The Ultimate guide, Sumitabha Das, Mc.Graw Hill.

Semester –III to VI

Skill Enhancement Courses (SEC-A & B): Choices : Semesters-3 to 6		
Courses	**Topics**	**Credit**
CMS-G-SEC-A-X-1-TH	Communication, Computer Network and Internet	02
CMS-G-SEC-A-X-2-TH	Software Engineering	02
CMS-G-SEC-B-X-1-TH	Multimedia and its Applications	02
CMS-G-SEC-B-X-2-TH	Information Security	02

CMS-G-SEC-A-X-1-TH: Communication, Computer Network and Internet
Skill Enhancement Course – A (SEC-A-1): Choice-1: Theory: 40 hours

Communication and Computer Network: (30 hours)

Introduction: Components, Uses, Application
Network Hierarchy: LAN, MAN, WAN; Topology;
Reference Model: OSI; Functionalities of each layer, **Data and Signals (Analog and Digital):** Periodic & Non-periodic signals, Bandwidth, Bit Rate, Baud Rate, Bit Length, and Composite Signal.

Transmission Media: Transmission Spectrum, Guided (Twisted Pair, Coaxial, Optical Fiber) and Unguided (Radio Wave, Microwave, Infrared, and Satellite Communication: Geostationary, Low Orbit and VSAT), Noise, Attenuation.

Digital Transmission: Line Coding (NRZ, RZ, Manchester); Block Coding (Basic Idea); Code Modulation (PCM, DM), Concepts of ADSL Modem.

Analog Transmission: Shift Keying (ASK, FSK, PSK, QAM)

Multiplexing: FDM, TDM, WDM.

Internet: (10 hours)
Bridges, Routers, Modem, Connectivity concept, DNS, URL, ISDN, WWW, Browser, Protocols, TCP, IP Address, E-mail: Architecture and services, Voice and Video conferencing, Internet service providers, ADSL.

Text/ Reference Books:
1. Data Communication and Networking, B.A. Forouzan, TMH.
2. Data and Computer Communication, W. Stallings, Pearson Education.
3. Computer Network, Tanenbaum, Pearson Education.

CMS-G-SEC-A-X-2-TH: Software Engineering
Skill Enhancement Course – A (SEC-A-2): Choice-2: Theory: 40 hours

Introduction: (12 hours)
Defining System, open and closed system, modeling of system, Communication system,

Software life cycle, Different Models: Classical and Iterative Waterfall Model; Spiral Model; Prototype Model; Evolutionary Model and its importance towards application for different system representations, Comparative Studies

Software Requirement and Specification Analysis: (07 hours)
Requirements Principles and its analysis principles; Specification Principles and its representations

Software Design Analysis: (12 hours)
Different levels of DFD Design, Physical and Logical DFD, Use and Conversions between them, Decision Tables and Trees, Coupling and Cohesion of the different modules, COCOMO

Software Testing: (07 hours)
Software Verification and Validation; Testing objectives, Testing Principles, Testability; Error and Faults; Unit Testing, White Box and Blank Box Testing.

Software Quality Assurances: (02 hours)
Concepts of Quality, Quality Control, Quality Assurance

Text/ Reference Books:
1. Fundamentals of Software Engineering, Rajib Mall, PHI.
2. Software Engineering, Pressman.

CMS-G-SEC-B-X-1-TH: Multimedia and its Applications
Skill Enhancement Course – B (SEC-B-1): Choice-1: Theory: 40 hours

Multimedia System: (10 hours)
An overview of multimedia system and media streams, Source representation and compression techniques text, speech and audio, still image and video.

Multi-modal Communication: (10 hours)
Video conferencing, networking support.

Multimedia OS: (20 hours)
Synchronization and QoS, Multimedia Servers.

Text/ Reference Books:
1**.** Multimedia: Making it work, Tay Vaughan, TMH.
2.Multimedia: Computing, Communications Applications, R Steinmetz and K Naharstedt, Pearson.

CMS-G-SEC-B-X-2-TH: Information Security
Skill Enhancement Course – B (SEC-B-2): Choice-2: Theory: 40 hours

Overview (05 hours)
Overview of Security Parameters: Confidentiality, Integrity and availability-security violation, OSI security architecture.

Cryptography (15 hours)

Mathematical Tools for Cryptography, Symmetric Encryption Algorithm, Theory of Block cipher design, Risk assessment, Network security management, Firewalls, Web and wireless security management, Computer security log management, IT security infrastructure, Operating system security, user security, program security

Finite Field and Number Theory: (05 hours)

Groups, Rings, Fields-Modular, Prime numbers, Fermat's and Euler's Theorem

Internet Firewalls for Trusted System: (05 hours)

Roles of Firewalls, Firewall related terminology, Types of Firewalls.

E-Mail, IP & Web Security (Qualitative study) (10hours)

E-mail Security: Security Services for E-mail-attacks possible through E-mail.

IP Security: Overview of IPSec, IP Security Architecture, Authentication Header, Encapsulation Security Payload.

Web Security: Secure Socket Layer/Transport Layer Security, Basic Protocol, SSL Attacks, Secure Electronic Transaction (SET).

Text/ Reference Books:

1. M. Bishop, ”Computer Security: Art and Science”, Pearson Education, 2003.
2. M. Stamp, ”Information Security: Principles and Practice”, John Wiley & Sons, 2005.
3. Cryptography and Network Security, William Stallings, Eastern Economy Edition, PHI.

Semester – V & VI

Discipline Specific Elective Courses (DSE-A & B): Choices: Semesters-5&6

Discipline Specific Elective- A (DSE- A): Candidate has to opt any 2 from the following topics		
CMS-G-DSE-A-5-1-TH	Data base Management System (DBMS)	04
CMS-G-DSE-A-5-1-P	DBMS Lab using SQL	02
CMS-G-DSE-A-5-2-TH	Operation Research	04
CMS-G-DSE-A-5-2-P	Operation Research Lab using C	02
CMS-G-DSE-A-5-3-TH	Computer Graphics	04
CMS-G-DSE-A-5-3-P	Computer Graphics Lab using C	02
Discipline Specific Elective- B (DSE- B): Candidate has to opt any 2 from the following topics		
CMS-G-DSE-B-6-1-TH	Embedded Systems	04
CMS-G-DSE-B-6-1-P	Embedded Systems Lab.	02
CMS-G-DSE-B-6-2-TH	Object Oriented Programming	04
CMS-G-DSE-B-6-2-P	Object Oriented Programming by Java	02
CMS-G-DSE-B-6-3-TH	Computational Mathematics	04
CMS-G-DSE-B-6-3-P	Computational Mathematics Lab using C	02

CMS-G-DSE-A-5-1-TH: Database Management System
Discipline Specific Elective Course – A (DSE-A-1): Choice-1: Theory: 60 hours

Introduction: (12 hours)
Drawbacks of Legacy System; Advantages of DBMS; Layered Architecture of Database, Data Independence; Data Models; Schemas and Instances; Database Languages.

ER Model: (12 hours)
Entity, Attributes and Relationship; Structural Constraints; Keys; ER Diagram of Some Example Database; Weak and Strong Entity Set; Symbolic Conventions; Specialization and Generalization; Constraints of Specialization and Generalization; Aggregation.

Relational Model: (14 hours)
Basic Concepts of Relational Model; Relational Algebra; Tuple Relational Calculus

Relational Database Design: (22 hours)
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.

CMS-G-DSE-A-5-1-P: DBMS Lab using SQL
Discipline Specific Elective Course – A (DSE-A-1): Choice-1: Practical: 40 hours

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); Derived Relations.

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.

CMS-G-DSE-A-5-2-TH: Object Oriented Programming
Discipline Specific Elective Course – A (DSE-A-2): Choice-2: Theory: 60 hours

Concept of OOPs (02 hours)
Difference with procedure oriented programming, Data abstraction and information hiding: Objects, Classes, methods.

Introduction to Java (04 hours)
Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods).

Arrays, Strings and I/O (08 hours)
Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

Object-Oriented Programming Overview (04 hours)
Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata. (14 hours)
Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

Exception Handling, Threading, Networking and Database Connectivity (15 hours)
Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using

java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Applets (13 hours)

Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

CMS-G-DSE-A-5-2-P: Object Oriented Programming by Java
Discipline Specific Elective Course – A (DSE-A-2): Choice-2: Practical: 40 hours

Object Oriented Programming Lab. by using Java

Text/Reference Books

1. Java: The Complete Reference, Herbert Schildt, McGraw-Hill Education.
2. The Java Language Specification, Java SE by James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley, Published by Addison Wesley.
3. Effective Java by Joshua Bloch, Publisher: Addison-Wesley.
4. Core Java 2 by Cay S. Horstmann, GaryCornell, Volume 1 , Prentice Hall.
5. Programming with Java by E. Balaguruswamy, McGraw Hill.
6. Java: How to Program by Paul Deitel, Harvey Deitel, Prentice Hall.
7. Programming with JAVA by John R. Hubbard, Schaum's Series.

CMS-G-DSE-A-5-3-TH: Computer Graphics
Discipline Specific Elective Course – A (DSE-A-3): Choice-3: Theory: 60 hours

Introduction (05 hours)

Basic concepts of Graphics Devices– CRT monitor, Monochrome and Color Monitor displaying technique only, Physical and logical units of graphics devices – Pixel and its different properties, Basic idea for image or picture formation using pixels – Raster Scan and Vector Scan.

Basic geometrical shapes formation algorithms (05 hours)

Concepts Co-ordinate System, Line Segment, Digital Differential Analyzer, Circle and arc segment, Bresenham's and Midpoint scan conversion algorithms.

Two Dimensional Transformations (14 hours)

Transformations operations - Translation, Rotation, Scaling. Reflection, Shearing and Inverse of these operations, Homogeneous coordinate system representation, matrix representation.
Composite Transformations Operations – Basic ideas and matrix representations by matrix concatenation for a particular operation.

Two Dimensional Clipping (08 hours)

View port, window port, display device, Point Clipping, Line Clipping, Cohen-Sutherland line clipping algorithm, Sutherland Hudgeman polygon clipping algorithm

Projection (08 hours)

Basic Concept of Projection operation and its application, Classification – Perspective, Parallel and its subclasses, Principles of these projections (Geometric representation only, no Mathematical Foundation and algorithms)

Applications (02 hours)

Basic Concepts Computer Art, Animation – Animating and modeling of real world, Morphing – Classification of morphing and Application to the Advertisements and publicities.

CMS-A-DSE-A-5-3-P: Computer Graphics Lab using C
DSE-A: Choice-3: Practical: 02 Credit: 40 hours

Computer Graphics lab is only based on theory including only Two-dimensional Transformation and Line Drawing.

Text/ Reference Books:

1. Computer Graphics by Zhigang Xiang, Roy Plastock, Schaum's Outlines Series.
2. Computer Graphics by Hearn & Baker, Pearson.
3. Procedural Elements for Computer Graphics by David F. Roger, 2nd Edition, TMH.
4. Computer Graphics by Foley, Van Dam, Feimer& John, Pearson.
5. Introduction to Computer Graphics and Multimedia, Mukhophadhyay and Chattopadhyay, Vikas publication.

CMS-G-DSE-B-6-1-TH: Embedded Systems
Discipline Specific Elective Course – B (DSE-B-1): Choice-1: Theory: 60 hours

Introduction to 8051: (10 hours)

Overview of Microcontroller, Memory, I/O interface
Intel Microcontroller 8051: Architecture, Peripheral Interface Controller (PIC).

Assembly Language Programming: (10 hours)

Instruction set, Addressing Modes, Jump, Loop and Call instructions, I/O Manipulation, Serial communication, Arithmetic and logical instructions.

Introduction to Embedded System Programming: (20 hours)

Data types and time delays, I/O programming, Logic operations, Data conversions, Data serialization, Interrupt programming, LCD and Keyboard interfacing, ADC, DAC, sensors interfacing, interfacing 8255, I/O interfacing for 8051, interfacing 8255, 8257, 8259/ 8279, ADC, DAC.

Hardware Description Language (VHDL): (20 hours)

Basic Terminology, Entity Declaration, Architecture body, Configuration and package declaration, Package body, Model analysis and Simulation.

CMS-A-DSE-B-6-1-P: Embedded Systems Lab.
DSE-A: Choice-3: Practical: 02 Credit: 40 hours

Practical: Sample practical problems can be included related to theory.
1. Assembly Language Programming related to Microcontroller 8051.
2. VHDL programs for construction and simulation of various digital circuits.

Text/ Reference Books:
1. David E.Simon, "An Embedded software primer", Pearson Education.
2. Raj Kamal, "Embedded Systems:, TMH.
3. Raj Kamal, "Microcontroller", Pearson Education.
4. A VHDL Primer, J. Bhasker, Prentice Hall

CMS-G-DSE-B-6-2-TH: Operation Research
Discipline Specific Elective Course – B (DSE-B-2): Choice-2: Theory: 60 hours

Introduction: (05 hours)
Origin and development of operation research, Nature and characteristic features, models in O.R.

Linear Programming Problem: (05 hours)
Introduction, mathematical formulation of the problem.

Simplex Method: (20 hours)
Introduction, computational procedure, artificial variable, problem of degeneracy.

Duality: (10 hours)
Concept, formulation of primal – dual, duality and simplex method, Dual Simplex method.

Transportation Problem (05 hours)
Introduction, mathematical formulation, finding initial basic feasible solution, optimality, degeneracy.

Game Theory: (10 hours)
Some basic terminology, Two-person Zero-sum Game, Game without Saddle Point – Mixed strategy, Algebraic method for 2×2 Game

Assignment Problem: (05 hours)
Introduction, mathematical formulation and solution.

CMS-A-DSE-B-6-2-P: Operation Research (O.R.) Lab. using C/ Python
DSE-B: Choice-2: Practical: 02 Credit: 40 hours
Lab sessions related to Simplex Method, Transportation Problem and Assignment Problem.

Text/ Reference Books:
1. Operations Research by Kanti Swarup, P.K. Gupta, Man Mohan, Sultan Chand & Sons
2. Schaum's Outline of Operations Research, Richard Bronson and Govindasami Naadimuthu, McGraw-Hill Education
3. Operations Research: An Introduction, Hamady.A. Taha, TMH

CMS-G-DSE-B-6-3-TH: Computational Mathematics
Discipline Specific Elective Course – B (DSE-B-3): Choice-3: Theory: 60 hours

Errors: (05 hours)
Introduction, Types of errors

Interpolation: (05 hours)
Newton's Forward and Backward Interpolation.

System of Linear Equations: (10 hours)
Properties: linear dependency, Rank, Singularity of coefficient matrix,
Solution methods: Gaussian Elimination, Gauss-Jordan Elimination.

Solution of Non-linear Equations: (10 hours)
Bisection algorithm, Newton-Raphson method.

Integration: (10 hours)
Trapezoidal and Simpson's $1/3^{rd}$ Rules and their composite forms

Graph Theory: (concept only) (20 hours)
Basic Terminology, Models and Types, Multi graphs and Weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees.

CMS-G-DSE-B-6-3-P: Computational Mathematics Lab.
Discipline Specific Elective Course – B (DSE-B-3): Choice-3: Practical: 40 hours

Lab. based on the Graph theory and Numerical Methods using C.

Text/ Reference Books:

1. Numerical Analysis and Computational Procedures by Mollah; New Central Book.
2. Computer Oriented Numerical Methods, 3rd Edition, V Rajaraman, PHI
3. Graph Theory With Applications To Engineering And Computer Science by Narsingh Deo, PHI.
4. Introduction to Graph Theory by D B West, 2nd edition, Pearson Education

UNIVERSITY OF CALCUTTA

Notification No.CSR/20/2023

It is notified for information of all concerned that in terms of the provisions of Section 54 of the Calcutta University Act, 1979, (as amended), and, in exercise of her powers under 9(6) of the said Act, the Vice-Chancellor has, by an order dated 25.07.2023 approved the syllabus of the under mentioned subjects semester wise Four-year (Honours & Honours with Research) /Three-year (Multidisciplinary) programme of U.G. courses of studies, as applicable under CCF,2022, under this University, as laid down in the accompanying pamphlet.

1. Food & Nutrition
2. Defence Studies
3. Human Development
4. Sanskrit (Revised syllabus after incorporating some amendments, in the syllabus published in CSR/18/23,dt.24.7.23)
5. Persian
6. Computer Science
7. B.Mus.(Honours) and Music (Minor)
8. Marks distribution for semesters 1 & 2 and Amendments in the syllabus of English (4-year Honours/3-year MDC)
9. Amendments in SEC paper of Physiology (Honours & Honours with Research) Courses (as mentioned in CSR/13/23,11.07.2023)
10. Environmental Science (Revised syllabus after incorporating some amendments, in the syllabus published in CSR/18/23, DT.24.7.2023)

The above shall take effect from the academic session 2023-2024.

SENATE HOUSE

Kolkata-700073

The 28th July ,2023

28/7/2023

Prof.(Dr.) Debasis Das

Registrar

SEM	CORE SUBJECTS (DSCC)		SEC
	PAPER NAME	PRACTICAL DETAILS	
1	Computer Fundamentals & Digital Logic (4)	Logic Design using ICs (basic circuits, focus on combinational part)	Data Visualization using Spreadsheets (4)
2	Problem Solving Using C (4)	Problem solving using C Lab (using gcc compiler)	Web Development (4) (HTML, PHP)
3	Data Structures (4)	Upto BSTs (using C)	Mobile App Development (4) (using Android Studio)
	Computer Architecture & Organization (4)	Logic Design using ICs (building logic blocks, & sequential ckts)	
4	Computational Mathematics (4)	Numerical Methods (using C)	**N.A.**
	Microprocessor (4)	8085 MPU Programming	
	Operating System (4)	Shell Programming (including system calls)	
	Object Oriented Programming (4)	Java Lab	
5	Design & Analysis of Algorithms (4)	Graph algorithms (using C++)	**N.A.**
	Data Communication and Networking (4)	Tutorial	
	Theory of Computation (4)	Tutorial	
	Database Management System (DBMS) (4)	MySQL & JS	
6	Software Engineering (4)	Tutorial (System Analysis & Design Lab)	**N.A.**
	Programming in Python (4)	Python Lab	
	Linear Algebra & Statistical Methods (4)	Related to theory (using Python)	
7	Compiler Design (4)	Tutorial	**N.A.**
	Machine Learning (4)	Related to theory (using Python)	
	Computer Graphics (4)	Using Python	
	IoT & Embedded Systems (4)	IoT & Embedded Systems (Python and IoT, Embedded Hardware)	
	Big Data Analytics / Research Project (4)	Big Data Analytics (Hadoop, MongoDB, Java Spark)	
8	Digital Image Processing (4)	Python with OpenCV	**N.A.**
	Cryptography (4)		
	Data Warehousing (4)		
	Mobile & Wireless Computing/Research Project (4)	Mobile & Wireless Computing (Network Simulation)	
	Cloud Computing (4) / Project	Cloud Computing (Using Cloud Simulator Learning Virtualisation and Developing Cloud Services)	

University of Calcutta

B.Sc (Honours and Honours with Research) 4 - years degree program in Computer Science under credit framework.

(2023)

Semester – I & II

Semester - I				
Paper	**Paper type**	**Paper name**	**Credit**	**Contact hours**
DSC/CC-1	Theory	Computer fundamentals and Digital Logic	3	45
	Practical	Computer fundamentals and Digital Logic lab	1	30
SEC – 1	Theory	Data visualization using spreadsheet	3	45
	Practical	Data visualization using spreadsheet Lab	1	30

Semester - II				
Paper	**Paper type**	**Paper name**	**Credit**	**Contact hours**
DSC/CC-2	Theory	Problem Solving using C	3	45
	Practical	Problem Solving using C Lab	1	30
SEC – 2	Theory	Web Development	3	45
	Practical	Web Development Lab	1	30

Semester - I				
Paper	Paper type	Paper name	Credit	Contact hours
DSC/CC-1	Theory	Computer fundamentals and Digital Logic	3	45
	Practical	Computer fundamentals and Digital Logic lab	1	30
SEC – 1	Theory	Data visualization using spreadsheet	3	45
	Practical	Data visualization using spreadsheet Lab	1	30

CMSA- Theory: Computer Fundamentals and Digital Logic
Core Course, Theory, Semester – 1, Credits - 03, Contact hours - 45.

Course description:

The course introduces the fundamental principles and concepts of digital logic, which form the foundation of digital systems and computer architecture. Students will learn about Boolean algebra, logic gates, combinational and sequential circuits, and the design and analysis of digital systems.

Course Objectives:

By the end of the course, students should be able to:

1. Understanding of Computer fundamentals, generations, classification of computers and brief understanding of languages used.
2. Understand the principles and terminology of digital logic.
3. Analyze and simplify Boolean expressions using Boolean algebra.
4. Design and implement combinational logic circuits using logic gates.
5. Design and analyze sequential logic circuits, including flip-flops and registers.
6. Apply digital logic concepts to solve practical problems.
7. Utilizing discrete logic gates and integrated circuits on breadboards for the design of digital circuits to enhance hands-on experience and practical understanding.

Computer Fundamentals	
Central Processing Unit (CPU), Primary memory and Secondary Storage devices, I/O devices, generation and classification of Computers: Super, Mainframe, Mini and Personal Computer, System and Application Software, basic concepts on machine, assembly and high level language.	2 hours
Number Systems	
Weighted and Non - Weighted Codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Gray Codes, Alphanumeric codes, ASCII, EBCDIC, Conversion of bases, signed arithmetic, 1's, 2's complement representation, Parity bits. **Single bit error detection and correcting codes:** Hamming Code. **Fixed and floating point Arithmetic**.	3 hours
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), XOR and others, Minterm, Maxterm, Minimization of Boolean Functions using Karnaugh-Map up to four (4)	4 hours

variables, two level and multilevel implementation using logic gates, simplification of logic expressions.	
Combinational Circuits	
Adder & Subtractor: Half adders (2-bit), half Subtractor (2-bit), Full Adder (3-bit), Full Subtractor (3-bit) realization using logic gates, Carry Look Ahead adders, BCD adder, 1's and 2's complement adders/subtractor unit using 4-bit parallel adders.	5 hours
Data Selector/Multiplexer: Realization of multiplexers (4 to 1 and 8 to 1) using logical gates, expansion (Cascading), realization of AND, OR and NOT using multiplexers, realization of different Boolean expressions (SOP) using multiplexers.	5 hours
Data Distributor: De-multiplexer, Cascading, realization of various functions.	2 hours
Encoders: Realization of simple and priority encoders using basic and universal logic gates.	2 hours
Chip Selector/Minterm Generator: Realization of decoders using logic gates, function realization, BCD Decoders, Seven Segment display and decoders, cascading.	3 hours
Parity bit, Code Converters and magnitude comparators: Parity bit generator/checker, Gray to binary code, binary to Gray code and Gray to Excess-3 code converter, 2 & 3 bit magnitude comparators.	2 hours
Sequential Circuits	
Latch & Flip-Flops: Basic Set/Reset (SR) Latch using NAND and NOR gates, Gated S-R latches, Gated D Latch, Gated J-K Latch, race around condition, Master-Slave J-K flip flop, negative and positive clock edge detector circuits, edge triggered SR, D, JK, and T flip flop, flip-flop Conversions.	5 hours
Registers: Serial Input Serial Output (SISO), Serial Input Parallel Output (SIPO), Parallel input Serial Output (PISO), Parallel Input Parallel Output (PIPO), Universal Shift Registers.	3 hours
Counters: **Asynchronous Counter** UP/DOWN Counters, Mod - N Counters, BCD Counter (Counter Construction using J-K and T Flip Flops).	4 hours
Synchronous Counter: UP/DOWN Counters, Mod-N Counters, Ring & Johnson Counters.	3 hours
Integrated Circuits (Qualitative Study): DTL, **TTL:** Concepts of Fan in & out, TTL NOT, TTL NAND & NOR, NMOS, PMOS, CMOS, **IC fabrication (Concepts only):** SSI, MSI, LSI, VLSI, ULSI.	2 hours

Core Course/DSE, CMSA- Practical: Computer Fundamentals and Digital Logic Lab, Semester – 1, Credits - 01, Contact hours - 30.

Combinational Circuits

1. Study and prove De-Morgan's Theorem.
2. Realization of Universal functions using NAND and NOR gates.
3. Implementation different functions (SOP, POS) using digital logic gates.
4. Implementation of half (2-bit) and full adder (3-bit) using basic (AND, OR and NOT) and Universal logic gates (NAND & NOR).
5. Design 4 to 1 multiplexer using basic or Universal logic gates and implement half and full adder/subtractor.
6. Design and implement half and full adder/subtractor and other functions using multiplexers 74151/74153 and other necessary logic gates.
7. Cascading of Multiplexers.
8. Design 2 to 4 decoder using basic or universal logic gates, study 74138 or 74139 and implement half and full Adder/Subtractor and other functions.
9. Design a display unit using Common anode or cathode seven segment display and decoders (7446/7447/7448)
10. Design and implement 4-input 3-output (one output as valid input indicator) priority encoder using basic (AND, OR & NOT) logic gates.
11. Design a parity generator and checker using basic logic gates.

Sequential Circuits

1. Realization of SR, D, JK Clocked/Gated, Level Triggered flip-flop using logic gates.
2. Master Slave flip-flop using discrete digital logic gates.
3. Conversion of flip-flops: D to JK, JK to D, JK to T, SR to JK, SR to D Flip-flop.
4. Design asynchronous counters MOD-n (upto 4 bits) UP/ DOWN.
5. Construction Synchronous UP/Down Counter (maximum 4 bits).

Note: 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.

Recommended Books

1. Digital Fundamentals, 11th Edition by Pearson Eleventh Edition, Thomas L. Floyd.
2. Digital Logic and Computer Design, M Morris Mano, Pearson.
3. Digital Electronics, Principles, Devices and Applications, Anil K. Maini, John Wiley & sons.
4. Digital Principles and Applications, Leach, Malvino, Saha, Tata McGraw Hill Education.
5. Digital Systems, Principal and Applications, Widmer, Moss and Tocci, Pearson.

CMSA- Theory: Data visualization using spreadsheet
SEC-1, Theory, Semester – 1, Credits - 03, Contact hours - 45.

Course Description

This Skill Enhancement Course (SEC) provides a comprehensive introduction to essential concepts and practical skills required for proficient utilization of spreadsheets. Students will gain proficiency in data management, visualization, analysis, and presentation using a widely-used open source spreadsheet software application such as Open Office, Libre Office, or Google Spreadsheets. Through this course, students will acquire the ability to proficiently create, format, manipulate, and analyze data within spreadsheets to meet a diverse range of needs.

Course Objectives

1. The purpose and potential applications of spreadsheets.
2. Create, format, and modify spreadsheets.
3. Use of formulas, functions, and calculations to perform data visualization.
4. Understanding and utilization of advanced spreadsheet features such as data validation, conditional formatting, and pivot tables.
5. Design visually appealing charts and graphs to represent data.
6. Collaborate and share spreadsheets with others.
7. Apply spreadsheet skills to real-world scenarios and problem-solving.
8. Role of spreadsheets in data analysis.
9. Import, clean, and transform data for analysis.
10. Applicability of statistical and mathematical functions for data visualization.
11. Advanced features and tools for data visualization.
12. Perform exploratory data analysis and identify patterns and trends.
13. Create informative reports and summaries based on data analysis.
14. Apply data analysis techniques to real-world problems.

Description	Teaching hours
Introduction to Spreadsheets Spreadsheets and their applications, overview of spreadsheet software (e.g., Open office, Google Sheets, Excel), creating workbooks, modifying workbook, modifying workbook, zooming in on a worksheet, arranging multiple workbook windows, adding buttons to the quick access toolbar, customizing the ribbon, maximizing usable space in the program window navigating the spreadsheet interface, entering and editing data in cells saving, opening, and closing spreadsheet files.	2 hours
Working with Data and Tables Entering and revising data, moving data within a workbook, finding and replacing data, correcting and expanding upon worksheet data, defining tables.	2 hours
Performing Calculations on Data Naming groups of data, creating formulas to calculate values (e.g., SUM, AVERAGE, COUNT), summarizing data that meets specific conditions (e.g., AVERAGEIF, COUNTA, COUNTBLANK, COUNTIFS, SUMIF, IFERROR etc), finding and correcting errors in calculations.	2 hours

Changing Workbook Appearance Formatting Cells, defining styles, workbook themes and table styles, making numbers easier to read, changing the appearance of data based on its value, adding images to worksheets.	2 hours
Data Analysis and Manipulation Limiting data appearance on screen, working with text functions for data cleaning, Splitting and combining data, Data normalization and standardization, working with ranges and named ranges, conditional formatting, data validation and error checking, using logical functions (e.g., IF, AND, OR), sorting and filtering data.	4 hours
Advanced Spreadsheet Features Creating and managing tables, creating and modifying pivot tables, using lookup functions (e.g., VLOOKUP, HLOOKUP), working with charts and graphs, importing and exporting data.	4 hours
Statistical Functions and Analysis Descriptive statistics (mean, median, mode, variance, etc.), Calculating measures of central tendency and dispersion, Correlation and regression analysis, Hypothesis testing and confidence intervals, Analysis of variance (ANOVA).	5 hours
Pivot Tables and Data Aggregation Creating pivot tables for data summarization, grouping and aggregating data by categories, Applying filters and slicers to pivot tables, calculating calculated fields and items.	4 hours
Advanced Data Visualization Creating charts and graphs for data representation, Customizing chart elements (titles, axes, legends), Using sparklines and data bars for visual analysis, Creating interactive dashboards, Incorporating trendlines and forecasting in charts.	5 hours
Exploratory Data Analysis Identifying patterns and outliers in data, Creating histograms and box plots, Using conditional formatting for data visualization, Data segmentation and drill-down analysis, Applying data validation rules for data integrity.	4 hours
Advanced Analysis Techniques Using goal seek and solver for optimization problems, Performing "what-if" analysis with data tables, Simulating data using random number functions, Monte Carlo simulation for risk analysis, creating scenario analysis models.	4 hours
Reporting and Presentation of Results Designing informative reports and summaries, creating interactive dashboards for data presentation, data visualization best practices, documenting data analysis processes presenting findings to stakeholders.	3 hours
Collaboration and Sharing Protecting worksheets and workbooks, sharing spreadsheets with others, tracking changes and commenting, collaborating in real-time, using version history and revision control.	4 hours

CMSA- Practical - Data visualization using spreadsheet
SEC, Laboratory, Semester – 1, Credits - 01, Contact hours - 30.

1. Create a personal budget spreadsheet that tracks income, expenses, and savings over a specified period. Use formulas and functions to calculate totals, percentages, and remaining balances.

2. A dataset containing sales data for a company to be provided. A spreadsheet to be created that calculates monthly sales totals, identifies top-selling products, and visualizes sales trends using line charts or bar graphs. Use conditional formatting to highlight exceptional sales performances.

3. Design a grade book spreadsheet that calculates students' final grades based on assignments, exams, and participation. Incorporate weighted grading systems, formulas for calculating averages, and conditional formatting to indicate performance levels. Generate reports to track individual student progress.

4. Create a spreadsheet that tracks inventory for a hypothetical business. Include columns for item names, quantities, prices, and total values. Use formulas to automatically update inventory totals, generate alerts for low stock, and create visualizations to represent inventory levels over time.

5. Loan parameters, such as principal amount, interest rate, and loan term to be provided. Create a spreadsheet that calculates monthly loan payments, remaining balances, and interest paid over time using appropriate formulas. Create a chart to visualize the loan's repayment schedule.

6. Dataset to be provided which will allow various data analysis tasks using spreadsheets. Calculation of summary statistics, sorting and filtering data, creating pivot tables for deeper insights, and generation of charts or graphs to visualize patterns or trends within the data.

7. A dataset to be selected (e.g., stock prices, weather data, population growth, etc) and create line charts or area charts to visualize trends over time. Students should choose appropriate chart types, label axes, and add titles and legends to make the visualization clear and informative.

8. A dataset containing information about different products or variables (e.g., sales data, customer satisfaction ratings) to be provided and following to be done; create bar charts or column charts to compare the performance or rankings of the items. Use color, data labels, and chart elements to enhance the visual comparison.

9. A dataset containing time-series data for multiple variables (e.g., monthly sales data for different products) to be provided and the following task to be performed; to create a combo chart with lines and columns to compare the trends of the variables and identify any relationships or patterns.

10. To create a unique visualization using advanced spreadsheet features and tools. For example, an experiment with sparklines, radar charts, or treemaps to represent specific types of data or explore innovative ways to visualize information.

Note: 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.

Recommended Text books

1. Data Analysis and Decision Making with Microsoft Excel" by S. Christian Albright.
2. Microsoft Excel 2019 Data Analysis and Business Modeling, Sixth Edition, Wayne L. Winston, Pearson education.
3. Excel 2019 Bible, Michael Alexander, 11th edition, Wiley.
4. Microsoft Office 2019 for Dummies, Wallace Wang, Wiley.

Recommended Application Software

1. Google Spreadsheets
2. Libre/Open Office
3. Excel sptreadsheets

Semester - II				
Paper	**Paper type**	**Paper name**	**Credit**	**Contact hours**
DSC/CC-2	Theory	Problem Solving using C	3	45
	Practical	Problem Solving using C Lab	1	30
SEC – 2	Theory	Web Development	3	45
	Practical	Web Development Lab	1	30

CMSA- Theory: Problem Solving using C
DSC/CC-2, Theory, Semester – 2, Credits - 03, Contact hours - 45.

Objective of the Course

The objectives of this course are to make the student understand programming language, programming, concepts of Loops, reading a set of Data, stepwise refinement, Functions, Control structure, Arrays. After completion of this course the student is expected to analyze the real life problem and write a program in 'C' language to solve the problem. The main emphasis of the course will be on problem solving aspect i.e. developing proper algorithms.

After completion of the course the student will be able to;

1. Develop efficient algorithms for solving a problem.
2. Use the various constructs of a programming language viz. conditional, iteration and recursion.
3. Implement the algorithms in "C" language.
4. Use simple data structures like arrays, stacks and linked list in solving problems.
5. Handling File in "C".

Outline of Course

S. No.	Topic	Minimum number of hours
1	Introduction to Programming	03
2	Algorithm/ Flowchart for Problem Solving	06
3	Introduction to 'C' Language	02
4	Conditional Statements and Loops	05
5	Arrays	05
6	Functions	04
7	Storage Classes	02
8	Structures and Unions	05
9	Pointers	05
10	Self-Referential Structures and Linked Lists	04
11	File Processing	02
12	Organizing C Projects	02
Lectures = 45 Practical/tutorials = 30, Total = 75		

Detailed Syllabus

Description	Teaching hours
Introduction to Programming The Basic Model of Computation, Algorithms, Flow-charts, Programming Languages, Compiler, Interpreter, Assembler, Linker and Loader, Testing and Debugging, Documentation.	03 hours
Algorithms/ Flowchart for Problem Solving Exchanging values of two variables, summation of a set of numbers, decimal base to binary base conversion, reversing digits of an integer, GCD (Greatest Common Division) of two numbers, test whether a number is prime, organize numbers in ascending order using bubble sort, find integer square root of a number, factorial computation, Fibonacci sequence, evaluate 'sin x' as sum of a series, reverse order of elements of an array, find largest number in an array, print elements of upper triangular matrix, multiplication of two matrices, evaluate a Polynomial.	06 hours
Introduction to 'C' Language Character set, variables, identifiers and their nomenclature, built-in data types, variable declaration, arithmetic operators and expressions, constants and literals, simple assignment statement, basic input/output statement, simple 'C' programs.	02 hours
Conditional Statements and Loops Decision making within a program, conditions, relational operators, logical connectives, if statement, if-else statement, **Loops:** while loop, do while, for loop, nested structure, infinite loops, switch-case, break, continue statement, structured programming.	05 hours
Arrays One dimensional arrays: Array manipulation; Searching, Insertion, deletion of an element from an array; finding the largest/smallest element in an array; two dimensional arrays, addition/multiplication of two matrices, Transpose of a square matrix; null terminated strings as array of characters, standard library string functions.	05 hours
Functions Top-down approach of problem solving, modular programming and functions, standard library of C functions, Prototype of a function: Formal parameter list, return type, function call, block structure, passing arguments to a function: call by reference, call by value, Recursive functions, arrays as function arguments.	04 hours
Storage Classes Scope and extent, Storage Classes in a single source file: auto, extern and static, register, Storage Classes in a multiple source files: extern and static	02 hours
Structures and Unions Structure variables, initialization, structure assignment, nested structure, structures and functions, structures and arrays: arrays of structures, structures containing arrays, unions	05 hours
Pointers Address operators, pointer type declaration, pointer assignment, pointer initialization, pointer arithmetic, functions and pointers, Array of Pointers, pointer to an array, pointers and structures, dynamic memory allocation.	05 hours

Self-Referential Structures and Linked Lists Creation of a singly connected linked list, Traversing a linked list, Insertion into a linked list, Deletion from a linked list	04 hours
File Processing Concept of Files, File opening in various modes and closing of a file, Reading from a file, Writing onto a file, Appending to a file.	02 hours
Organizing C projects, working with multiple source directories, makefiles.	02 hours

Recommended books main reading

1. Byron S Gottfried "Programming with C" Second edition, Tata McGraw Hill, 2007 (Paperback)
2. R.G. Dromey, "How to solve it by Computer", Pearson Education, 2008.
3. Kanetkar Y, "Let us C", BPB Publications, 2007.
4. Hanly J R & Koffman E.B, "Problem Solving and Program design in C", Pearson Education, 2009.
5. Kashi Nath Dey and Samir Bandyopadhayay "C Programming Essentials" Pearson India Education, 2010.

Supplementary reading.

1. E. Balagurusamy, "Programming with ANSI-C", Fourth Edition,2008, Tata McGraw Hill.
2. Venugopal K. R and Prasad S. R, "Mastering 'C'", Third Edition, 2008, Tata McGraw Hill.
3. B.W. Kernighan & D. M. Ritchie, "The C Programming Language", Second Edition, 2001, Pearson education.
4. ISRD Group, "Programming and Problem-Solving Using C", Tata McGraw Hill,2008.
5. Pradip Dey, Manas Ghosh, "Programming in C", Oxford University Press, 2007.

CMSA- Practical: Problem Solving using C
DSC/CC-2, Practical, Semester – 2, Credits - 01, Contact hours - 30.

Algorithms / Flowchart (Sample and simple assignments)

1. Design a flowchart/ Algorithm for a basic calculator that accepts two numbers and an operator (+, -, *, /) as input from the user and performs the corresponding operations, and displaying/print the result.
2. Create a flowchart/Algorithm that converts a temperature from Celsius to Fahrenheit or vice versa based on user input.
3. Design a flowchart/Algorithm that calculates the factorial of a given positive integer provided by the user.
4. Create a flowchart/Algorithm that finds and displays the largest number among three input numbers given by the user.
5. Design a flowchart/Algorithm to implement the linear search algorithm to find a specific element in an array of integers. The array and the element to search for should be taken as user input.
6. Create a flowchart/Algorithm that calculates the area and perimeter/circumference of different shapes (e.g., circle, rectangle, triangle) based on user input for dimensions.
7. Design a flowchart/Algorithm that checks whether a given input string is a palindrome or not.

Introduction to 'C' Language (Assignments/examples related to simple C program.)

8. Write a program in C to read two numbers and produce the sum and product of those numbers and show the result separately.
9. Write a program in C to read two numbers and print the greater number, if both the numbers are same them print "EQUAL".
10. Write a program in C multiple numbers say n and print the greatest and the third greatest.
11. Write a program in C to read n numbers and print the even/odd numbers up to n.
12. Write a program in C to read a number and print the sum of n natural numbers.
13. Write a program in C to read a number n and print factor of n.
14. Write a program in C to read a number n and print first 10 multiples of n.
15. Write a program in C to read a number n and print if n is "PRIME" or "COMPOSITE".
16. Write a program in C to calculate the average of a set of N numbers.
17. Write a program in C convert the temperature given in Celsius to Fahrenheit or vice-versa.
18. Write a program in C to determine and print the sum of the following harmonic series for a given value of n: 1+1/2+1/3+……..1/n.
19. Write a program in C that reads a floating-point number and then displays the right most digits of integral part of the number.
20. Write a program in C to accept the length and breadth in meters and calculate the area and perimeter and also determine if it is a rectangle or a square based on the inputs given.
21. Write a program in C to accept an input and determine if the input entered is a number or alphabet or a special character.
22. Write a program in C to accept a word and then print the reverse case that is lower to upper or upper to lower case.
23. Write an interactive program in C which will demonstrate the process of division/multiplication, the user should be asked to enter two-digit numbers.

Conditional Statements and Loops (simple examples)

24. Write a program in C to read a number n and print n terms of the Fibonacci series.
25. Write a program in C to read a number n and print a single digit answer showing sum of the digits of n. (example – input 8626, expected output – 4, explanation 8+6+2+6 = 22, 2+2 = 4).
26. Write a program in C to read a number n and print all the prime numbers up to n.
27. Write a program in C to read a number n and print the following pattern (input = 5, expected output
    1
    12
    123
    1234
    12345).
28. Write a program in C to check if the given number is the Armstrong number or not (e.g 153 = 13+53+33).
29. Write a program in C to check the type of the given triangle whether it is equilateral, isosceles or scalene.

Arrays (examples of few simple programs)

30. Write a program in C to read a string and store it into a character array. Check whether the string is a palindrome or not and display accordingly.

31. Write a program in C to read a list of numbers stored in an integer array and while saving them arrange in ascending order.
32. Write a program in C to read two matrices and perform addition.
33. Write a program in C to read two matrix and check their compatibility for multiplication, if compatible then find product and print it.
34. Write a program in C to read a string and print the triangular pattern using the string.

Functions

35. Write a program in C to print all the Armstrong number from 1 to 500.
36. Write a function ***convert ()*** that returns a weight in Kg after being given a weight in pounds.
37. Write a function to find all perfect numbers from 1 to 100 (perfect numbers are positive integers where the sum of perfect divisor is the number itself, e.g., 6 = 1+2+3).
38. Write a function power () to find base raise to power [$\mathbf{base}^{\mathbf{power}}$].
39. Write a program in C to find solution of a quadratic equation [$x = \frac{-b \pm \sqrt{b^2-4a}}{2a}$] where values a, b and c to be accepted from the user as input.
40. Accept inputs from the user and echo it on to the screen in normal as well as in reverse using void recursive function.
41. Accept any number from the user and calculate the factorial of the number using recursion
42. Accept numbers n and print the odd/even numbers up to n using recursive function.
43. Write a program in C in compute the cubes of all numbers from 10 to 20.
44. Write a program in C to find the GCD of a number.
45. Write a program in C to generate all combinations of 1, 2, 3, 4 using recursion, e.g.,1234, 2341….. etc.

Storage Classes

46. Write a program in C to accept a number and find the factorial of the number demonstrating use of automatic variables.
47. Write a program in C to accept two numbers and find the sum of the number demonstrating use of external variables.
48. Write a program in C to accept two numbers and find the sum of the number demonstrating use of global variables.
49. Write a program in C to illustrate the use of static variables.
50. Write a program in C to accept numbers till a negative number is entered and calculate the sum of a list of numbers read using static variable.
51. Write a program in C to sum integers and use static variable to store the cumulative sum.

Pointers

52. Write a program in C to swap two numbers of n length.
53. Write a program in C for swapping numbers using functions.
54. Write a program in C to illustrate the Call by Value and Call by reference a rule in C programming.
55. Write a program in C to use a double dimensional array and print each cells value and address.
56. Write a program in C to show the use of Array, declared at compilation time (static manner) to read 10 numbers and display them.
57. Write a program in C to show the use of Array, declared dynamically to read 10 numbers and display them.
58. Write a program in C to read a string in a dynamic array and determine whether it is palindrome or not.

Structures and Unions

59. Write a program in C to read the data of a student, store it in a structure and display it.
60. Write a program in C to read the data of many students, store it in a structure and display the student's data and average percentage of the class.
61. Write a program in C to accept two dates from the user, validate both of them and check if they are different dates.
62. Write a program in C to accept students' data from the user. Check if the student stream is science, commerce or arts. If the stream is arts, then print the class of students. If the stream is science, then print the grade and if the stream is commerce, then print the percentage.

Files

63. Write a program in C showing the technique of opening and closing a file say **result.dat** and writing a list of numbers and its square into the file.
64. Write some texts into a file, reopen the file in read mode and reproduce the text on the monitor (use of putc() and fputc()).
65. Write a few numbers in the file created earlier. Reopen it in Read mode, write odd numbers in one file and even number in another file (use the **getw** and **putw** functions).
66. Write programs to demonstrate the use of getc(), fgetc() and ungetc().
67. Write programs to demonstrate the use of String I/O, Formatted I/O and End of file eof() and feof().

Recommended assignment content/structure

- Objective
- Algorithm/Flowchart
- Code
- Result
- Conclusion

Platform/Compiler

- GCC

Note: 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.

CMSA- Theory: Web development
SEC, Theory, Semester – 2, Credits - 03, Contact hours - 45.

Course Description

This course provides an introduction to web development using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). Students will learn the core concepts and practical skills needed to create and style web pages. The course covers the fundamentals of HTML structure, CSS styling properties, and responsive web design principles.

Course Objectives

1. Understanding the basics of web development and the role of HTML and CSS.
2. Create well-structured HTML documents using proper tags and elements.

3. Apply CSS to style web pages, including layout, typography, colors, and images.
4. Implement responsive design techniques to ensure optimal display on different devices.
5. Incorporate multimedia elements, such as images, videos, and audio, into web pages.
6. Understand best practices for organizing and maintaining code in web development projects.
7. Develop and deploy a basic website using HTML and CSS.

Description	Teaching hours
Introduction to Web development Overview of web technologies and the role of HTML and CSS, understanding the structure of a web page, introduction to web browsers and developer tools.	3 hours
HTML Fundamentals Introduction to HTML tags and elements, creating headings, paragraphs, lists, and links, working with images and multimedia content, creating forms for user input.	3 hours
CSS basics Introduction to CSS and its role in web page styling, selectors, properties, and values, applying inline, internal, and external style sheets, formatting text, backgrounds, and borders.	3 hours
CSS Layout and box model Understanding the box model and its impact on layout, working with margins, padding, and borders, positioning elements using floats, positioning properties, and flexbox, creating responsive layouts with media queries.	3 hours
Typography and colors Styling text with fonts, sizes, weights, and styles, formatting text using CSS properties, understanding color models and applying colors to elements.	4 hours
Images and multimedia Working with images: sizing, aligning, and optimizing, incorporating videos and audio into web pages, implementing responsive images and media.	4 hours
CSS Selectors and specificity Understanding CSS selectors and specificity, applying styles to specific elements and classes, using pseudo-classes and pseudo-elements.	5 hours
Responsive Web design Introduction to responsive design principles, creating fluid layouts using CSS media queries, adapting web pages for different screen sizes and devices.	4 hours
CSS Frameworks and libraries Overview of popular CSS frameworks (e.g., Bootstrap, Foundation), using pre-built CSS components and grids, customizing and integrating CSS frameworks into web projects.	5 hours
Web development best practices Organizing and structuring code files and directories, validating HTML and CSS code, optimizing web pages for performance, introduction to version control with Git.	3 hours
Building and deploying a website Planning and designing a basic website structure, Implementing HTML and CSS to create the website, testing and debugging the website across different browsers, deploying the website to a local host/web server	6 hours

CMSA- Web development
SEC, Laboratory, Semester – 2, Credits - 01, Contact hours - 30.

1. Creating a personal portfolio website using HTML and CSS. There should be sections for an about me, projects, skills, and contact information's. Using CSS to style the layout, typography, and colors to create a visually appealing and professional-looking portfolio.

2. To design a responsive website that adapts to different screen sizes. They should create a layout that adjusts fluidly using CSS media queries and responsive design techniques.

3. To create a product landing page for a fictional product or an existing one. HTML to be used to structure the page and CSS to style the layout, typography, buttons, and images. Main focus to be on creating an engaging page that effectively showcases the chosen product.

4. To incorporate CSS animation effects into a web page. Use CSS transitions, transforms, and keyframe animations to add interactive and engaging elements to the website. Create animations for hover effects, scrolling effects, image sliders, or menu transitions.

5. Redesign an existing website using HTML and CSS. Analyze the original design and propose improvements to the layout, typography, color scheme, and overall user experience.

6. Create a webpage layout using CSS Flexbox or CSS Grid. Design a responsive layout that organizes content in a visually appealing way. Experiment can be performed with different grid or flexbox properties to create flexible and responsive designs.

7. To design and style an interactive form using HTML and CSS. They should incorporate various form elements such as text inputs, checkboxes, radio buttons, and select dropdowns. Apply CSS styling to improve the form's visual appearance and user experience.

Note: 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.

Suggested Readings.

1. Mastering HTML, CSS & Java Script Web Publishing, Laura Lemay, Rafe Colburn, Jennifer Kyrnin, BPB Publication.
2. Web designing and development, Satish Jain, BPB Publications.
3. HTML & CSS: The complete reference, Thomas Powell, McGraw Hill education.
4. Web programming with HTML5, CSS and JavaScript, John Dean, Joneas and Bartlet learning.
5. Sams Teach Yourself HTML, CSS, and JavaScript All in One, Julie C Meloni, Pearson Education.
6. Learning Web App development, Semmy Purewal, O'Reilly.

INTERDISCIPLINARY COURSE

Fundamentals of Computer Science and its applications 45 hrs

Course Outcome:

- Demonstrate the basic concepts of Computer science, such as Computer Architecture, Data representation, Algorithms, and Data structures.
- Write basic programs in a high-level programming language, such as Python.
- Explain how computers communicate with each other over a network.
- Explain how artificial intelligence is used in real-world applications.
- Use ICT tools to create documents, spreadsheets, and presentations.

Detailed Syllabus

- Introduction to computers and computing 08 hr.

History of computing and the different types of computers that are available today, Generations of computers, Basic Building blocks (CPU, Memory, I/O Devices), types of computer (Mainframe, Desktop, Laptop, System on Chip). Classification of Software – System and Application Software, Basic Security Anti-Virus.

- Data representation and number systems 04 hr

Concept of binary code, ASCII and how it is used to represent data in computers, How different number systems work

- Algorithms and data structures 06 hr

Basic concepts of algorithms and data structures: Common algorithms and data structures, such as sorting algorithms and linked lists.

- Office suite 08 hr

  Word processors, Spreadsheets, and Presentation

- Programming languages 08 hr

Basic concepts of programming languages: types of programming languages , machine language, assembly language, high level language, Introduction to writing basic programs in Python (Finding prime numbers, finding GCD of two numbers etc,)

- Networking 05 hr

Basic concept of networking and how computers communicate with each other, LAN, WAN, Introduction to the concept of the internet and how it works. Mobile communication

- Artificial intelligence 05 hr

Basic concept of artificial intelligence and how it is used in computers. Introduction to Machine Learning, Preliminary concept of Big Data, Recommendation System, Conversation Agents like ChatGPT, Prompt Engineering

- Information and Communications (ICT) Tools 01 hr

Importance of ICT tools, different types of ICT tools and their uses

Recommended Books:

1. Computer Science: An Interdisciplinary Approach, Robert Sedgewick (Author), Kevin Wayne (Author)
2. Introduction To Computer Science, Anita Goel Pearson India

Structure of Core Courses in Computer Science for Three-year MDC

Semester	Course / Paper Code	Course Name
1	CMS- MD- CC1- 1- Th/ P	Computer Fundamentals and Digital Logic
2	CMS- MD- CC2- 2- Th / P	Problem Solving using C

Structure of Minor Courses in Computer Science for MDC

Semester	Course / Paper Code	Course Name
3	CMS- MD- MC1- 3- Th / P	Computer Fundamentals and Digital Logic
4	CMS- MD- MC2- 4- Th / P	Problem Solving using C

Structure of Skill Enhancement Courses in Computer Science for MDC

Semester	Course / Paper Code	Course Name
Semester 1/2/3	CMS- MD- SEC2- 2- Th / P	Web Development (HTML / PHP)

Structure of Skill Interdisciplinary Courses in Computer Science for MDC

Semester	Course / Paper Code	Course Name
Semester 1/2/3	CMS- MD- IDC2- 2- Th / P	Fundamentals of Computer Science and their Applications