UNIVERSITY OF CALCUTTA

Notification No. CSR/ 81 /18

It is notified for information of all concerned that the Syndicate in its meeting held on 06.08.2018 (vide Item No. 08) approved some amendments in the existing notification (Notification No.CSR /12/18 dated 04.06.2018) pertaining to 1st and 2nd Semester Syllabus of B.Sc. Computer Science (Honours) under CBCS under this University, as laid down in the accompanying pamphlet.

The above shall be effective from the 2018-2019.

20/08/18

(Debabrata Manna)
Deputy Registrar (Acting)

SENATE HOUSE
KOLKATA-700073
The 20th August, 2018

Syllabus for B.Sc. (Honours) in Computer Science (CMSA) with Choice Based Credit System (CBCS) for Semesters–I&II from the Academic Session 2018-19

SEMESTER – I

Semester	*Courses*	*Topics*	*Credit*
I	**CMS-A-CC-1-1-TH (Core Course-1) Theory**	**Digital Logic**	**4**
	CMS-A-CC-1-1-P (Core Course-1) Practical	**Digital Circuits**	**2**
	CMS-A-CC-1-2-TH (Core Course-2) Theory	**Programming Fundamentals using C**	**4**
	CMS-A-CC-1-2-P (Core Course-2) Practical	**Programming in C**	**2**

SEMESTER – I

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

Introduction to Computer Fundamentals: (02 hours)
CPU, Primary and Secondary Storage, I/O Devices, Concept of Super, Mainframe, Mini and Personal Computer, System and Application Software (concept only).

Number Systems: (05 hours)
Weighted and Non-Weighted Codes, positional, Binary, Octal, Hexadecimal, Binary coded Decimal (BCD), Gray Codes, Alphanumeric codes, ASCII, EBCDIC, Conversion of bases, Parity bits, Single Error bit detection and correcting codes: Hamming Codes, Fixed and Floating Point Arithmetic: Addition, Subtraction, Multiplication and Division.

Boolean Algebra: (08 hours)
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 variables, Two level and multilevel implementation using logic gates, Simplification of logic expression.

Combinational Circuits: (20 hours)
Half adders, Full Adder (3-bit), Half Subtractor, Full Subtractor (3-bit) and construction using Basic Logic Gates (OR, AND, NOT) and Universal Logic Gates (NAND & NOR), Multibit Adder- Ripple Carry Adder, Carry Look Ahead adder, BCD Adder, 1'S & 2'S Complement Adder/Subtractor unit Construction using 4 bit Full adders units, 1 bit, 2 bit, 3 bit and 4 bit Comparators using basic logic gates.
Data Selector-Multiplexer: Expansion (Cascading), Reduction, 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.
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: (21 hours)
Set/Reset (SR) Latch: Using NAND and NOR gates, Gated S-R latches, D Latch, J-K Latch, T Flip Flop, Race around Condition, Master Slave J-K Flip Flop, Clock - Duty Cycle, Rising Time, Falling Time, Negative Edge Detector and Positive Edge Detector circuits, Edge Triggered SR, D and JK Flip Flop, Flip-Flop Conversions, Flip-Flops with Preset and Clear.
Registers: Serial Input Serial Output, Serial Input Parallel Output, Parallel input Serial Output, Parallel Input parallel Output, Universal Shift Registers.
Counters: Asynchronous Counter: UP/DOWN Counters, Mod - N Counters, BCD Counter (Counter Construction using J-K and T Flip Flops).
Synchronous Counter: UP/DOWN Counters, Mod-N Counters, Ring Counters, Johnson Counters.

Integrated Circuits (Concept only): (04 hours)
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

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

Combinational Circuits:

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

Sequential Circuits:

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

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, Mc Graw 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: (04 hours)
History, Basic Structure, Algorithms, Structured programming constructs.

C Programming elements: (08 hours)
Character sets, Keywords, Constants, Variables, Data Types, Operators- Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional, Operator Precedence and Associations; Expressions, type casting. Comments, Functions, Storage Classes, Bit manipulation, Input and output.

C Preprocessor: (06 hours)
File inclusion, Macro substitution.

Statements: (06hours)
Assignment, Control statements- if, if else, switch, break, continue, goto, Loops-while, do_while, for.

Functions: (06 hours)
Argument passing, return statement, return values and their types, recursion

Arrays: (07hours)
String handling with arrays, String handling functions.

Pointers: (10 hours)
Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation.

User defined Data types: (07 hours)
Enumerated data types, Structures. Structure arrays, Pointers to Functions and Structures, Unions

File Access: (06hours)
Opening, Closing, I/O operations.

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.

Computer Science (Honours) CMSA -CBCS Syllabus

SEMESTER – II

Semester	*Courses*	*Topics*	*Credit*
II	**CMS-A-CC-2-3-TH (Core Course – 3) Theory**	**Data Structure**	**4**
	CMS-A-CC-2-3-P (Core Course – 3) Practical	**Data Structure Lab.**	**2**
	CMS-A-CC-2-4-TH (Core Course – 4) Theory	**Basic Electronic Devices and Circuits**	**4**
	CMS-A-CC-2-4-P (Core Course – 4) Practical	**Basic Electronic Devices and Circuits Lab.**	**2**

SEMESTER – II

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

Introduction to Data Structure: (01 hour)
Abstract Data Type.

Arrays: (05 hours)
1D, 2D and Multi-dimensional Arrays, Sparse Matrices. Polynomial representation (Polynomial Representation as Application).

Linked Lists: (09 hours)
Singly, Doubly and Circular Lists; Normal and Circular representation of Self Organizing Lists; Skip Lists, Polynomial representation (Polynomial Representation as Application).

Stacks: (05 hours)
Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack

Queues: (05 hours)
Array and Linked representation of Queue, Circular Queue, De-queue, Priority Queues

Recursion: (05 hours)
Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

Trees: (15 hours)
Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees).

Searching and Sorting: (10 hours)

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Merge Sort, Quick sort, Shell Sort, Heap sort, Radix sort, Comparison of Sorting Techniques

Hashing: (05 hours)

Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collision by Open Addressing, Coalesced Hashing, Separate Chaining, Choosing a Hash Function, Perfect Hashing Function.

CMS-A-CC-2-3-P: Data Structure Lab.

Core Course- 3: Practical: 02 Credits: 40 hours

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates 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 using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. 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. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to create a Binary Search Tree and include following operations in tree:
    - **(a)** Insertion (Recursive and Iterative Implementation)
    - **(b)** Deletion by copying
    - **(c)** Deletion by Merging
    - **(d)** Search a no. in BST
    - **(e)** Display its preorder, postorder and inorder traversals Recursively
    - **(f)** Display its preorder, postorder and inorder traversals Iteratively
    - **(g)** Display its level-by-level traversals
    - **(h)** Count the non-leaf nodes and leaf nodes
    - **(i)** Display height of tree
    - **(j)** Create a mirror image of tree
    - **(k)** Check whether two BSTs are equal or not
12. WAP to reverse the order of the elements in the stack using additional stack.
13. WAP to reverse the order of the elements in the stack using additional Queue.
14. WAP to implement Diagonal Matrix using one-dimensional array.
15. WAP to implement Lower Triangular Matrix using one-dimensional array.
16. WAP to implement Upper Triangular Matrix using one-dimensional array.
17. WAP to implement Symmetric Matrix using one-dimensional array.

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, Silicon Pr.
2) Data Structures: A Pseudocode Approach with C, Richard F. Gilberg and Behrouz A. Forouzan, Cengage Learning
3) Data Structures In C, Noel Kalicharan, CreateSpace Independent Publishing Platform.
4) Adam Drozdek, Data Structures and algorithm in C, Cengage Learning.
5) The C Programming Language, Brian W. Kernighan and Dennis Ritchie, Prentice Hall.
6) Sartaj Sahni, Data Structures, Algorithms and applications in C++, Second Edition, Universities Press, 2011.
7) Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, Dat,a Structures Using C and C++, 2nd ed., PHI, 2009.

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

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: Working Principle, construction and characteristics of PN junction diode, biasing, depletion region, Single Phase Half, Full wave and bridge rectifier using PN Junction diode, Circuit, Working principle, Calculation of Average DC current and Voltage, RMS, Ripple Factor, efficiency, Peak Inverse Voltage (PIV). Zener diode: Characteristics and its application as a voltage regulator	(09 hours)
Bipolar Junction Transistor: Principle of Junction Transistor (including current components, current gains), Types: CE, CB, CC), DC biasing in CE mode: Q-Point, load line analysis, Transistor as an amplifier. Inverter using transistors: 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 (NOT, NAND, NOR)	(08 hours)
PNPN Devices: Working Principle of SCR, UJT, construction, characteristics and simple applications: SCR, DIAC, TRIAC, SCR regulated power supply, Switch Mode Power Supply (SMPS)	(08 hours)

qualitative study only. Concept and functions of Optoelectronic materials (LED, LCD, Photo Sensors and basics of Optical Fiber and Opto-couplers).

Operational Amplifiers (OPAMP): (12 hours)

Inverting Amplifier, Non-inverting Amplifier, Offset parameters, Inverting and Non-inverting Adder, Differentiator, Integrator, Scale changer and Schmitt Trigger. Concept of Virtual ground, CMRR, Signal Generation using OPAMP: Monostable, Astable (Square wave generator)

Timer: Construction and Functional description of 555, Mono-stable, Bistable and Astable Operation, VCO. (04 hours)

Data Acquisition: (04 hours)

R-2R ladder DAC, Weighted resistor type DAC, Flash Type ADC, Counter, Successive Approximation Register (SAR), Dual Slope ADC and Integrating Type.

CMS-A-CC-2-4-P: Basic Electronic Devices and Circuits Lab.

Core Course-4: Practical: 02 Credits: 40 hours

1. Study the forward characteristic of a p-n junction diode and calculate the static and dynamic resistance of the diode.
2. Construct a Full wave rectifier using power diodes and study its load regulation characteristics with or without capacitor filter.
3. Construct a Bridge rectifier using power diodes and study its load regulation characteristics with or without capacitor filter.
4. Construct a Zener Voltage regulator and study its load regulation characteristics.
5. Construct a positive and negative voltage regulator using Three terminal linear voltage regulator 78XX and 79XX. Study its load regulation characteristics.
6. 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.
7. Study the Output characteristics of a transistor in CE mode and calculate the gain from the graph.
8. Using Transistor to construct NOT or Invert Operation and draw the transfer characteristics and measure the threshold voltage.
9. Construct and study an Inverting Amplifier using OPAMP with different sets of inputs and feedback resistors and Calculate the gain from the graph.
10. Construct and study an Non-Inverting Amplifier using OPAMP with different sets of inputs and feedback resistors and Calculate the gain from the graph.
11. Construct and study an Inverting Adder using OPAMP.
12. Construct and study an Non-Inverting adder using OPAMP.
13. Construct and study a subtractor using OPAMP.
14. Construct and study the OPAMP as a differentiator.
15. Construct and study the OPAMP as a integrator.
16. Construct an Astable Multivibrator using Timer 555.
17. Study and construct a R-2R ladder digital to analog converter.
18. Convert an analog signal into digital using ADC 0809.

Text/Reference Books:

1. Electronic Devices & Circuits Theory, Boylested & Nashelsky, PHI.
2. Electronics fundamental & Application, Chattopadhyay, Rakshit, New Age International Publishers.
3. Op-Amps And Linear Integrated Circuits, R. A. Gayakwad, Prentice Hall.
4. Solid State Electronic Devices, Streetman, PHI.
5. Elements of Electronics, Bagde Singh, S Chand Publication.
6. Microelectronic circuits, Sedra Smith, Oxford.
7. Operational Amplifier and Linear Integrated Circuits, Coughlin Driscol.
8. Electronic Devices and Circuits, Salivahanan, Suresh Kumar, McGrawHill education