Proposed Syllabus For General Elective for students other than B.Sc. (Honours) in Computer Science Submitted

To

ENLIGHTENMENT TO PERFECTION

University of North Bengal
(Accredited by NAAC with grade A)

Under
Choice Based Credit System (CBCS)

[With effect from the Session 2018-19]

Basket-A

GE 1A: Digital Electronics [Credits : 4 Hours : 60]

Unit-1: Fundamentals of Computers (15 Lectures)

Generation of Computers and Computer Languages, Computer Systems, Basic block Diagram, Von- Neumann Architecture, Types of Computers, Hardware, Firmware, I/O Devices, Storage classifications, Language translators.

Unit-2: Logic gates (15 Lectures)

AND, OR, NOT Gates and their Truth Tables, NOR, NAND & XOR gates, Boolean algebra, Basic Boolean Laws, De-morgan‟s theorem, Boolean function and their truth tables, MAP simplification, Minimization techniques, K-Map, Sum of Product & Product of Sum, Venn diagram.

Unit-3: Circuits, Adders, Flip Flops, Registers etc. (15 Lectures)

Combinational & Sequential circuits, Half adder & Full adder, BCD adder, Full Subtractor, Flip-flops-RS, D, JK, T & Master-Slave flip-flops, Shift registers, Multiplexer, Encoder, Decoder.

Unit-4: Number Systems & Arithmetic (15 Lectures)

Number System: Positional, Binary, Octal, Decimal, Hexa-Decimal and their Representations. Methods of conversion from one Base to another. Binary Addition, Subtraction, Multiplication. Negative number representation: Sign magnitude, 1's, 2's Complement. Different coding schemes: BCD, EBCDIC, UNICODE, ASCII, GRAY, Excess-3 Codes. Fixed and floating point representation

Suggested Readings

1.Rajaraman V. & Radhakrishnan, An Introduction To Digital Computer Design, PHI.
2.Malvino & Leach, Digital Principles & Applications, TMH
3.S. Salivahanan, S. Arivazhagan, Digital Circuits and Design, Oxford University Press

GE 1B: Computer Networks [Credits : 4 Hours : 60]

Unit 1: Introduction to Computer Networks (10 Lectures)

Network definition; network topologies; network classifications; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.

Unit 2: Data Communication Fundamentals and Techniques (10 Lectures)

Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission; digital to analog modulation-; multiplexing techniques- FDM, TDM; transmission media.

Unit 3: Networks Switching Techniques and Access mechanisms (10 Lectures)

Circuit switching; packet switching- connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

Unit 4: Data Link Layer Functions and Protocol (10 Lectures)

Error detection and error correction techniques; data-link control- framing and flow control; error recovery protocols- stop and wait ARQ, go-back-n ARQ; Point to Point Protocol on Internet.

Unit 5: Multiple Access Protocol and Networks (8 Lectures)

CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways; Unit 6: Networks Layer Functions and Protocols (6 Lectures) Routing; routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols

Unit 7: Transport Layer Functions and Protocols (6 Lectures)

Transport services- error and flow control, Connection establishment and release- three way handshaking

Unit 8: Overview of Application layer protocol (6 Lectures)

Overview of DNS protocol; overview of WWW & HTTP protocol

Suggested Readings:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM, 2007.
2. A. S. Tanenbaum: Computer Networks, Fourth edition, PHI , 2002
3. Dr. Rakesh Kumar Mandal: Computer Networks for Students, First Edition, SPD, 2018

GE 2A: Programming in C [Credits : 4 Hours : 60]

Unit 1: Introduction to C (6 Lectures)

History of C, Overview of Procedural Programming,
Introduction to Algorithm & Flowcharts.

Unit 2: Understanding Compilation and Execution in C (6 Lectures)

Using main() function Compiling and Executing Simple Programs in C.

Unit 3: Data Types, Variables, Constants, Operators and Basic I/O (7 Lectures)

Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types,
Operators (Arithmetic, Logical and Bitwise), Using Comments in programs,
Character I/O (getc, getchar, putc, putcharetc),
Formatted and Console I/O (printf(), scanf(), cin, cout),
Using Basic Header Files (stdio.h, iostream.h, conio.hetc).

Unit 4: Expressions, Conditional Statements and Iterative Statements (7 Lectures)

Simple Expressions in C(including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct),
Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

Unit 5: Understanding Functions (6 Lectures)

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Unit 6: Implementation of Arrays and Strings (6 Lectures)

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays

Unit 7: User-defined Data Types (Structures and Unions) (4 Lectures)

Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

Unit 8: Pointers and References in C (8 Lectures)

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with Pointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values

Unit 9: Memory Allocation in C (3 Lectures)

Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation

Unit 10: File I/O (4 Lectures)

Opening and closing a file, Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files,

Unit 11: Preprocessor Directives (3 Lectures)

Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

Suggested Readings

1."The C Programming Language ANSI C Version", Kernighan & Ritchie, Prentice Hall Software Series

2."ANSI C - Made Easy", Herbert Schildt, Osborne McGraw-Hill

3."Learning to Program in C", N. Kantaris, Babani

4."C - The Complete Reference", Herbert Schildt, Osborne McGraw-Hill

5."Programming in C", Reema Thareja, Oxford University Press

6."A First Course in Programming With C", T. Jeyapoovan, Vikas Publishing House

7."Let Us C", Yashavant P. Kanetkar, BPB Publications

GE 2AL: Programming in C LAB [Credits : 2 Hours : 60]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. WAP to perform input/output of all basic data types.
2. WAP to enter two numbers and find their sum.
3. WAP to reverse a number.
4. WAP to Swap Two Numbers.
5. WAP to Check Whether a Number is Even or Odd
6. WAP to compute the factors of a given number.
7. WAP to enter marks of five subjects and calculate total, average and percentage.
8. WAP to print the sum and product of digits of an integer.
9. WAP to Check Whether a Character is Vowel or Consonant
10. WAP to Find the Largest Number Among Three Numbers
11. WAP to perform following actions on an array entered by the user:
    a) Print the even-valued elements
    b) Print the odd-valued elements
    c) Calculate and print the sum and average of the elements of array
    d) Print the maximum and minimum element of array
    e) Remove the duplicates from the array
    f) Print the array in reverse order
12. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
13. Write a program that swaps two numbers using pointers.
14. Write a program in which a function is passed address of two variables and then alter its contents.
15. 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.
16. Write a program to find sum of n elements entered by the user.
17. Write a menu driven program to perform following operations on strings:
18. Show address of each character in string
19. Concatenate two strings without using strcat function.
20. Concatenate two strings using strcat function.

GE 2B: Microprocessor [Credits : 4 Hours : 60]

Unit 1: Microprocessor architecture (20 Lectures)

Internal architecture, system bus architecture, memory and I/O interfaces

Unit 2: Microprocessor programming (20 Lectures)

Register Organization, instruction formats, assembly language programming

Unit 3: Interfacing (20 Lectures)

Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, display, timer, interrupt controller, DMA controller, video controllers, communication interfaces.

Suggested Readings

1.Barry B. Brey: The Intel Microprocessors: Architecture, Programming and Interfacing. Pearson Education,Sixth Edition, 2009.

2.Walter A Triebel, Avtar Singh; The 8088 and 8086 Microprocessors Programming, Interfacing, Software,Hardware, and Applications. PHI, Fourth Edition 2005.

3.Ramesh Gaonkar; Microprocessor Architecture, Programming, And Applications With The 8085 5/E,Penram International Publishing(India)

4. B. Ram, Fundamentals of Microprocessors and Microcontrollers, Dhanpat Rai Publications

GE 2BL: Microprocessor Lab [Credits : 2 Hours : 60]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems: ASSEMBLY LANGUAGE PROGRAMMING in 8085

1. Write an assembly language program to check whether a number is even or odd.
2. Write an assembly language program to find the number of ones and the number of zeros in an 8-bit number.
3. Write an assembly language program to find the smaller of two numbers.

4. Write an assembly language program to multiply two 8-bit numbers.
5. Write an assembly language program to find the smallest among 10 integers stored in memory locations starting from 2050H
6. Write an assembly language program to find the largest among 10 integers stored in memory locations starting from 2050H
7. Write an assembly language program to sort 10 numbers using bubble sort.
8. Write an assembly language program to generate the first 10 fibonacci series and store the result at memory location starting from FC50H
9. Write an assembly language program to find the sum of the series 12 + 22 + 32 + 42 + ….. + 102
10. Write an assembly language program to find the perfect square of any number and if the number is not a perfect square, display FFH
11. Write an assembly language program for linear search.
12. Write an assembly language program to calculate the following expression using a single register: Y = X2 + 2X + 3XZ
13. Write an assembly language program to find the sum of the first 10 even natural numbers.
14. Write an assembly language program to find the sum of the first n odd natural numbers.
15. Write an assembly language program to create an odd parity generator.
16. Write an assembly language program to create an even parity generator.
17. Write an assembly language program to find the sum of five 8-bit numbers.
18. Write an assembly language program to convert decimal to binary.
19. Write an assembly language program to convert octal to binary.
20. Write an assembly language program to convert hexadecimal to binary.
21. Write an assembly language program to convert hexadecimal to decimal.
22. Write an assembly language program to check whether an 8-bit number is palindrome or not.
23. Write an assembly language program to display the truth table for and AND gate.
24. Write an assembly language program to display the truth table for and OR gate.
25. Write an assembly language program to display the truth table for and XOR gate.
26. Write an assembly language program to perform n byte addition of two numbers.
27. Write an assembly language program to implement a simple sub routine call.
28. Write an assembly language program to check whether a number is prime or not.

Basket-B

GE 3A: Operating Systems [Credits : 4 Hours : 60]

Unit 1: Introduction (10 Lectures)

Basic OS functions, resource abstraction, types of operating systems– multiprogramming systems, batch systems , time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

Unit 2: Operating System Organization (6 Lectures)

Processor and user modes, kernels, system calls and system programs.

Unit 3: Process Management (20 Lectures)

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter-process communication; deadlocks.

Unit 4: Memory Management (10 Lectures)

Physical and virtual address space; memory allocation strategies -fixed and variable partitions, paging, segmentation, virtual memory

Unit 5: File and I/O Management (10 Lectures)

Directory structure, file operations, file allocation methods, device management.

Unit 6: Protection and Security (4 Lectures)

Policy mechanism, Authentication, Internal access Authorization.

Suggested Readings:

1.A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications2008.

2.A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.

3.G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.

4.W. Stallings, Operating Systems, Internals & Design Principles, 5th Edition, Prentice Hall of India. 2008.

5.M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

6.Operating Systems, A K Sharma, University Press

GE 3B: Database Management Systems [Credits : 4 Hours : 60]

Unit 1: Introduction (6 Lectures)

Characteristics of database approach, data models, database system architecture and data independence.

Unit 2: Entity Relationship(ER) Modeling (8 Lectures)

Entity types, relationships, constraints.

Unit 3: Relation data model (20 Lectures)

Relational model concepts, relational constraints, relational algebra, SQL queries .

Unit 4: Database design (15 Lectures)

Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms(up to BCNF).

Unit 5: Transaction Processing (3 Lectures)

ACID properties, concurrency control

Unit 6: File Structure and Indexing (8 Lectures)

Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files(Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Suggested Readings:

1.R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.

2.R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.

3.A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGraw Hill, 2010.

4.R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6thEdition, Pearson Education,2013.

5.Database Book: The Principles and Practices Using MySQL, NarainGehani, University Press

GE 4A: Programming in JAVA [Credits : 4 Hours : 60]

Unit 1: Introduction to Java (4 Lectures)

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)

Unit 2: Arrays, Strings and I/O (8 Lectures)

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.

Unit 3: Object-Oriented Programming Overview (4 Lectures)

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.

Unit 4: Inheritance, Interfaces, Packages, Enumerations, Auto boxing and Metadata (14 lectures)

Inheritance: (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, Auto boxing/Un boxing, Enumerations and Metadata.

Unit 5: Exception Handling, Threading, Networking and Database Connectivity (15 Lectures)

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.

Unit 6: Applets and Event Handling (15 Lectures)

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.

Suggested Readings:

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
2. James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley"The Java Language Specification, JavaSE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
3. Joshua Bloch, "Effective Java" 2nd Edition,Publisher: Addison-Wesley, 2008.
4. Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 1 ,9th Edition,Printice Hall.2012
5. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, PrinticeHall.2013
6. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
7. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.

GE 4AL: Programming in JAVA Lab [Credits : 2 Hours : 60]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .length in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBufferclasss like setCharAt(,setLength(), append(), insert(), concat()and equals().
9. Write a program to create a —distance class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer.
10. Modify the —distance class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword

13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacii series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program —Divide By Zero that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throw able).

GE 4B: Python Programming [Credits : 4 Lectures : 60]

Unit 1: Planning the Computer Program(12 Lectures)
Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation

Unit 2: Techniques of Problem Solving(12 Lectures)
Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top- down and bottom-up programming.

Unit 3: Overview of Programming(12 Lectures)
Structure of a Python Program, Elements of Python

Unit 4: Introduction to Python(12 Lectures)
Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator)

Unit 5: Creating Python Programs(12 Lectures)
Input and Output Statements, Control statements(Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments.

Suggested Readings:

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.or 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python ,Freely available online.2012
4. http://docs.python.org/3/tutorial/index.html
5. http://interactivepython.org/courselib/static/pythonds
6. http://www.ibiblio.org/g2swap/byteofpython/read/

GE 4BL Software Lab Based on Python: [Credits : 2 Hours : 60]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

Section: A (Simple programs)

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria :
   a. Grade A: Percentage >=80
   b. Grade B: Percentage>=70 and <80
   c. Grade C: Percentage>=60 and <70
   d. Grade D: Percentage>=40 and <60
   e. Grade E: Percentage<40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to implement the use of arrays in Python.
6. WAP to implement String Manipulation in python in Python.
7. WAP to find sum of the following series for n terms:
   1 – 2/2! + 3/3! - - - - - n/n!

Section: B (OOPs using Python):

All the programs should be written using user defined functions, wherever possible.

1. *WAP to create Class and Objects in Python.*
2. *WAP to implement Data Hiding in Python.*
3. *WAP to implement constructor and destructor for a class in Python.*
4. *WAP to implement constructor and destructor in Python.*
5. *WAP to implement different types of inheritance in Python.*
6. *WAP to implement concept of Overriding in Python.*
7. *Write programs to create mathematical 3D objects using class.*
   a. curve b. sphere c. cone d. arrow e. ring f. cylinder