ASSAM SCIENCE AND TECHNOLOGY UNIVERSITY

Guwahati

Course Structure and Syllabus

(From Academic Session 2018-19 onwards)

M.Sc. COMPUTER SCIENCE

1st Semester

ASSAM SCIENCE AND TECHNOLOGY UNIVERSITY

Guwahati

Course Structure and Syllabus

(From Academic Session 2018-19 onwards)

M.Sc. Computer Science 1st Semester

Semester I / M.Sc. Computer Science

Sl. No.	Sub-Code	Subject	Hours per Week			Credits
			L	T	P	C
Theory						
1	MCS182101	Operating System	3	1	0	4
2	MCS182102	Digital Logic and System Design	3	1	0	4
3	MCS182103	Computer Organization and Architecture	3	1	0	4
4	MCS182104	Programming Techniques-I (C, C++)	3	1	0	4
Practical						
1	MCS182114	Lab I: Programming Techniques-I (C,C++)	0	0	4	2
		Total	12	4	4	**18**
Total Contact Hours per week : 20						
Total Credit: 18						

Course Code	Course Title	Hours per week L-T-P	Credit C
MCS182101	**Operating System**	**3-1-0**	**4**

MODULE 1: Review of computer organization

Major subsystems, instruction sets, I/O organization.

MODULE 2: Memory architecture

Address protection, segmentation, virtual memory, paging, page replacement algorithms, cache memory, hierarchy of memory types, associative memory.

MODULE 3: Support for concurrent process

Mutual exclusion, shared data, critical sections, busy form of waiting, lock and unlock primitives, synchronization block and wakeup.

MODULE 4: Scheduling

Process states, virtual processors, interrupt mechanism, scheduling algorithms, implementation of concurrency primitive.

MODULE 5: System deadlock

Prevention, detection and avoidance.

MODULE 6: Multiprogramming system

Queue management, I/O supervisors, memory management. File system, disk and drum scheduling.

MODULE 7: Case Study

Some real operating system– semaphores, messages, shared memory.

MODULE 8: Advanced Topics

Secondary storage management, Security, Distributed operating system.

Reference Books:

1. A. S. Tanenbaum and A. S. Woodhull, "Operating Systems Design and Implementation" PHI
2. Stallings, Unix Network programming, PHI.
3. Kerninghan and Pike, The Unix programming Environment, PHI.
4. M. Bach, "The Design of the Unix Operating System", PHI
5. A. S. Tanenbaum, "Design of Operating System", Addison Wesley
6. J. L. Peterson and A. Silberschatz, "Operating System concepts" Addison – W

Course Code	Course Title	Hours per week L-T-P	Credit C
MCS182102	**Digital Logic and System Design**	**3-1-0**	**4**

MODULE 1: Electronic Devices

Passive and Active Components, Resistor, Capacitor, Inductor etc. Diode, Zener Diode, Diode and Zener Diode Characteristic, Diode as rectifier (Full wave and Half Wave), Concept of Power Supply, Trensistor-CE, CB Configuration, Transistor Characteristics, Baising of Transistor, Transistor as Switch.

MODULE 2: Representation of Information Number System

Binary, Octal, Hexadecimal, Positive and Negative Number, 1's and 2's complement, Arithmetic Operations: Addition, Subtraction, etc. Character codes: ASCII and BCD.
Error detection and correction, parity codes and Hamming codes, etc.

MODULE 3: Logic Design

Logic Gates and their electronic realization, Boolean Algebra, Boolean variables and functions-canonical and standard forms, minimization of Boolean functions – Karnaugh Map.

MODULE 4: Combinational Design

Implementation of Boolean function and logic gates, concept of combinational Design-Adder, Subtractor, Multiplexer, decoders, encoders, simple arithmetic and logic circuits.

MODULE 5: Sequential Design

Concept of latch, Clock, Study of Flip-Flop- S-R, J-K, D, T etc. counters- synchronous and asynchronous, Modes of counter, shift registers, etc.

MODULE 6: Digital ICs

Digital Logic families- Study the Characteristics like Fan-out, Fan –in etc of TTL CMOS logic, study of AND, OR, NOT in TTL and CMOS families.

MODULE 7: Memory Devices

Semiconductor memory- RAM, ROM, Magnetic core and surface memory-disk, drum, tape, Access time and cost considerations: concept and volatility, random access, serial access, direct access, online and backup storage, PAL, FPGA.

MODULE 8: Basic CPU Organization

Simple functional block diagram of a CPU, instruction execution process.

Reference Books:

1. 1.Millman & Halkisius Mc Graw Hill

2. Digital Principal and Application TMH
3. Digital Logic and Computer Design PHI
4. An introductory to computer design-PHI

Course Code	Course Title	Hours per week L-T-P	Credit C
MCS182103	**Computer Organization and Architecture**	**3-1-0**	**4**

MODULE 1: Introduction to Computers

Basic of Computer, Von Neumann Architecture, Generation of Computer, Classification of Computers, Instruction Execution.

MODULE 2: Register Transfer and Micro operations

Register Transfer, Bus and Memory Transfers, Tree-State Bus Buffers, Memory Transfer, Micro-Operations, Register Transfer Micro-Operations, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations.

MODULE 3: Computer Arithmetic

Addition and Subtraction with Signed-Magnitude, Multiplication Algorithm, Booth Multiplication Algorithm, Array Multiplier, Division Algorithm, Hardware Algorithm, Divide Overflow, Floating-Point Arithmetic Operations, Basic Considerations, Register Configuration, Addition and Subtraction, Decimal Arithmetic Operations, BCD Adder, BCD Subtraction.

MODULE 4: Programming the Basic Computer

Machine language, Assembly language, Assembler, first pass, second pass, Programming Arithmetic and Logic operations, Multiplication Program, Double Precision Addition, Logic operations, Shift operations.

MODULE 5: Organization of a Computer

Central Progressing Module (CPU), Stack Organization, Register Stack, Memory Stack, Reverse Polish Notation. Instruction Formats, Three- Address Instructions, two – Address Instructions, One- Address Instructions, Zero Address Instructions, RISC Instructions, Addressing Modes Reduced Instruction Set Computer, CISC Characteristics RISC Characteristics.

MODULE 6: Input-Output Organization

Modes of Transfer, Priority Interrupt, DMA, Input-Output Processor (IOP), CPUIOP Communication.

MODULE 7: Memory Organization

Memory Hierarchy, Main Memory, Auxiliary Memory, Cache Memory, Virtual Memory. Address Space and Memory Space, Associative Memory, Page Table, Page Replacement.

MODULE 8: Parallel Processing and Vector Processing

Pipelining, Parallel Processing, Pipelining General Consideration, Arithmetic Pipeline Instruction Pipeline, vector processors and array processors. Introduction to parallel processing, Inter processor communication & synchronization. Vector Operations, Matrix Multiplication, Memory Interleaving.

Reference Books:

1. M. Moris Mano, Computer System Architecture, Pearson Education.
2. William Stallings, Computer Organisation and Architecture, Pearson Education.
3. David A Patterson, Computer Architecture, Pearson Education.
4. P. Pal Choudhri, Computer Organisation and Design, PHI.
5. J. P. Hayes, Computer System Architecture, Pearson Education.
6. Kai Hawang, Advanced Computer Architecture, Tata McGraw Hill.

Course Code	Course Title	Hours per week L-T-P	Credit C
MCS182104	**Programming Techniques-I (C, C++)**	**3-1-0**	**4**

Using C

MODULE 1: An Overview

Problem Identification, analysis, design, coding, testing & debugging, implementation, modification & maintenance; algorithm & flowcharts; Characteristics of a good program – accuracy, simplicity, robustness, minimum resource & time requirement modularization; Rules / Conventions of coding, documentation, naming variables; Top-down and bottom-up design.

MODULE 2: Fundamentals of C Programming

History of C; Structure of a C Program; Data Types; Constant & Variables, naming variables; Operators & Expressions; Control Statements – if-else, for, while, do-while, Case switch statement; break, continue, exit (), go to & labels, Arrays; Formatted & unformatted I/O; Type modifiers & storage classes; Ternary operator; Type conversion & type casting; Priority & associativity of operators.

MODULE 3: Modular Programming

Functions; Arguments; Return Value; Parameter passing – call by value, call by reference; return statement; scope, visibility and life time rules for various types of variable, static variable; calling a function; Recursion – basics, comparison with iteration, types of recursion – direct, indirect, tree and tail recursion, when to avoid recursion, examples.

MODULE 4: Advanced Programming Techniques

String; Pointer v/s array; Pointer to pointer; Array of pointer & its limitation; Function returning pointers; Pointer to function, Function as parameter; Structure – basic, declaration, membership operator, pointer to structure, referential operator, self-referential structures, structure within structure, array in structure, array of structures; Union – basic, declaration; Enumerated data type; Type def. command line arguments.

MODULE 5: Miscellaneous Features

File handling and related function; printf & scanf family; C pre-processor – basics, #include, #define, #undef, conditional computation directive like #if, #else, #elif, #endif, #ifdef and #ifndef; Variable argument list functions.

Reference Books:

1. Kerninghan & Richied, “The C Programming Language”, PHI
2. Programming in Ansi C by E. Balaguruswamy, TMH, 2004
3. Let us C Yaswant Kanetkar, BPB publications
4. Gottfried: “Problem solving in C”, Schaum Series
5. How to solve it by Computer by R.G.Dromey (P.H.I) 1994

Using C++

MODULE 1: Principles of Object Oriented Programming

Object Oriented Programming Paradigm, Basic Concepts of Object Oriented Programming, Benefits of OOPs, Object Oriented Languages, Application of OOP, C++ statements, Class, Structures of C++, Program, Creating the Source File, Compiling and Linking.

MODULE 2: Tokens, Expressions and Control Structures

Introduction Tokens, Keywords, Identifiers, Basic Data Types, User Defined Data Types, derived Data Types, Symbolic Constants, Type Compatibility, Declaration of Variables, Dynamic Initialization of Variables, Reference Variables, Operators in C++, Scope Resolution Operator, Member Dereferencing Operators, Manipulators, Type Cast Operator, Expressions and Implicit Conversions, Operator Precedence, Control Structures.

MODULE 3: Class and Objects

Specifying a class, Defining Member Functions, making an Outside Function Inline, Nesting of Member Function, Private Member Function, Arrays within a class, Memory Allocation for Objects, Static Data Member, Static Member Functions, Array of Objects, Objects as Function Arguments.

MODULE 4: Constructors and Destructors

Introduction, Constructors, Parameterized Constructors, Multiple Constructors with Default Arguments, Dynamic Initialization of Objects, Copy Constructors, Dynamic Constructors and Destructors.

MODULE 5: Functions in C++

The Main Function, Function Prototyping, call by reference, Return by Reference, Inline Functions, Default Arguments, Constant Arguments, Function Overloading, Friend and Virtual Function.

MODULE 6: Operator Overloading and Type Conversions

Introduction, Defining Operator Overloading, Overloading Unary Operators, Overloading Binary Operators Using Friends, Manipulation of Strings using Operators, Rules for overloading operators, Type conversions

MODULE 7: Inheritance

Introduction, Defining Derived Classes, Single Inheritance, Making a Private Member Inheritable, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance.

MODULE 8: Polymorphism and Virtual Functions

Compile time Polymorphism, run time Polymorphism, Pointers to Objects, This Pointer, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions.

Text Books:

1. Object Oriented Programming with C++ by E. Balaguruswamy, TMH Publications
2. Object Oriented Programming in C++ by Nabajyoti Barakati SAMS PHI Pvt. Ltd.
3. Insights into OOPS & C++, Rajeshwar Shukla, Pragya Publications.

Course Code	Course Title	Hours per week L-T-P	Credit C
MCS182114	**Lab I: Programming Techniques-I (C, C++)**	**0-0-4**	**2**

Using C

List of Experiments: (Any 15 Experiments)

1. Write a program to find the sum of two numbers using function in C.
2. Write a program to find the largest of three numbers using C.
3. Write a program to find the roots of a quadratic equation using C.
4. Write a program to find the sum of given range of numbers using C.
5. Write a program to enter a number and then calculate the sum of digits using C.
6. Write a program to check a given number is palindrome or not using C.
7. Write a program to find the addition of two matrices using C.
8. Write a program to check whether a given string is palindrome or not using C.
9. Write a program to find the multiplication of two matrices using C.
10. Write a program to find the sum of diagonal elements of a matrix using C.
11. Write a program to merging the two arrays using C.
12. Write a program to find the factorial of a given number using C.
13. Write a program to generate the Fibonacci series up to 20 terms using C.
14. Write a program to swap the given tow numbers using c.
15. Write a program to determine whether an entered character is a vowel or consonant using C.
16. Write a program to take input from the user and then check whether it is a number or a character. If it is character, determine whether it is in uppercase or lowercase.
17. Write a program to classify a given number as prime or composite.
18. Write a program to enter a decimal number. Calculate and display the binary equivalent of this number using c.
19. Write a program using do-while loop to read the numbers until -1 is encountered. Count the number of prime numbers and composite numbers entered by the user.
20. Write a program to calculate the GCD of two numbers.
21. Write a program to sum of the series $1/1^2 + 1/2^2 + +\ldots\ldots\ldots + 1/n^2$
22. Write a program to find whether the given number is an Armstrong number or not.

Using C++

List of Experiments: (Any 10 Experiments)

1. Write a program to find the sum of two numbers using C++.
2. Write a program to convert Centigrade to Fahrenheit and vice versa using C++.
3. Write a program to find the larger of two numbers using C++.
4. Write a program to enter the code and price of items. The user must feed the quantity in which he wants a product. The program must calculate and display the final bill using C++.
5. Write a program to implement student's class. The program must demonstrates the use of nested member function, arrays within the class and making a function defined outside the class an inline function.
6. Write a program to enter a rational number, simplify and display it by using c++.
7. Write a program that adds two complex numbers. The object must be passed through reference and the result must be passed by value.
8. Write a program in c++ to display the list of students according to their ranks.
9. Write a program in c++ that uses an overloaded constructor to dynamically allocate memory to an array and thus find the largest of its elements.
10. Write a class that stores a string and all its status details such as number of upper case characters, vowels and consonant using c++.
11. Write a program using c++ to concatenate two strings. The memory for the strings must be allocated dynamically.
12. Write a program in c++ to add two binary numbers of four digits
13. Write a program using c++ to sort an array that has been allocated memory dynamically.
14. Write a program in c++ that define a class Employee and display the personal and salary details of five employees using single inheritance.
15. Define a class student with data member rollno and name. Derive a class Fees from student that has a data member fees and functions to submit fees and generate receipt. Derive another class Result from Student that displays the marks and grade obtained by the student.

****