Morigaon College (Autonomous)
Morigaon: Assam

SYLLABUS
Computer Application
(3rd Semester)

Computer Organization and Architecture

(Core/Credit: 4)

Course Code: BCA-CORE-MC-0300104

1. Learning Outcome: Student will be able to learn about the structure, function and characteristics of computer systems. Understand the design of the various functional units and components of computers. Identify the elements of modern instructions sets and their impact on processor design. Able to learn about identify and compare different methods for computer I/O. Student will able to learn about basics of assembly language.

2. Total Credit: 4, Theory: 4, Practical: 0

3. List of reference books:

   a) M.Morris Mano, Computer System Architecture, PHI publication.
   b) Hamachar, Vranesic and Zaky, Computer Architecture.
   c) William Stallings, Computer Organization and Architecture; Pearson.
   d) Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, 5th Edition.

4. Detailed Syllabus:

Unit 1: Introduction

Definitions of Computer Organization and Architecture, History of computer architecture, Basic functional blocks of a computer: CPU, memory, Input-output subsystems, Control unit, Types of register- general purpose registers, special purpose registers, index registers.

Unit 2: Data Representation

Number system, Complements, Representation of signed numbers, Subtraction of unsigned numbers, Fixed-Point representation- Integer representation, Arithmetic addition, Arithmetic subtraction, Overflow, Decimal Fixed-Point representation, Floating-Point representation, Other Binary Codes- Gray Code etc.

Unit 3: Register Transfer and Micro-operation

Introduction to Register Transfer Language, Register transfer, Bus and Memory transfers, Arithmetic micro-operation- Binary adder, Binary adder-subtractor, Binary incrementor, Arithmetic circuit, Logic micro-operation, Shift micro-operation, Arithmetic logic shift unit.

Unit4: Processing Unit

Instruction codes, Computer registers, General register organization, Register stack, Memory stack, Computer instructions, Data path in a CPU, Operations of a control unit, Hardwired control unit, Micro-programmed control unit, Instruction cycle, Operands, Addressing modes, Instruction format- Three-address instructions, Two-address instructions, One-address instructions, Zero- address instructions, Data transfer and manipulation- Data transfer instructions, Data manipulation instructions, Arithmetic instructions, Logical and Bit manipulation instructions, Shift instructions, Program Control-Status bit conditions, Conditional branch instructions, Subroutine call and return, Instruction execution cycle, CISC and RISC architectures.

Unit 5: Memory Organization

Semiconductor memories, Memory cells - SRAM and DRAM cells, Concept of hierarchical memory organization, Interleaved memories, Cache memory unit - Concept of cache memory, Mapping methods, Organization of a cache memory unit, Cache replacement policies, Write

policy, Concept of virtual memory.

Unit 6: I/O Organization

Access of I/O devices, I/O ports, I/O control mechanisms - Program controlled I/O, Interrupt driven I/O, DMA controlled I/O, Interrupts: Types of interrupts, Enabling and disabling interrupts, Handling interrupts.

Unit 7: Basics of Microprocessor and Assembly Language

Introduction to microprocessors, 8085 Microprocessor and its operation, 8085 instruction sets, Addressing modes in 8085, Classifications of instructions and addressing mode, Assembly language programming basics, Assembling, Executing and debugging the programs, Developing counters and Time delay routines, Interfacing concepts.

Mathematics II

(Core/Credit: 4)

Course Code: BCA-CORE-MC-0300204

1. Learning Outcomes: After successful completion of this course, students will be able to:
   a) Learn the basic concepts of limit, continuity and derivatives.
   b) Understand graphs and its different representations in Computers. How to model real life problems using graphs. Learn a few basic graph traversal algorithms.
   c) Understand the basic idea of counting and use it in counting under various constraints.
   d) Understand Mathematical Logic from algorithmic point of view.

2. Total Credit: 4, Theory: 4, Practical: 0

3. Reference books:
   (a) Discrete Mathematics structures with applications to Computer Science, J. P. Tremblay and R. Manohar, Mc-Graw Hill.
   (b) Discrete Mathematics, N. Ch.SN Iyengar, K.A. Venkatesh, V. M. Chandrasekaran, P. S. Arunachalam, Vikash Publishing House Pvt Ltd.
   (c) Elements of Discrete Mathematics, C. L. Liu, Mc-Graw Hill International Ed.

4. Course Details:

Unit1: Calculus

Intuitive idea of limits and continuity. Limits of polynomials and rational functions. Derivatives, Algebra of derivative of a function, Derivative of polynomials and trigonometric functions. Roll's theorem, Lagrange's Mean Value theorem and Taylor's theorem. Meaning of the sign of derivative, indeterminate forms, maxima and minima (single variable only).

Unit2: Graph theory

Basic Definition of graph, Directed, Undirected and Weighted Graphs. Representation of graphs in Computers – Adjacency Matrix and Adjacency Lists. Degree of vertices – indegree and outdegree. Paths, Cycles and Acyclic graphs. Simple operations on graphs and amount of computations required for each operation. Connected graph, Tree and Forest. Bipartite graph, Algorithms on graph traversals- Breadth first search, Depth first search.

Unit3: Combinatorics

Basic of counting principles, principle of inclusion-exclusion, application of inclusion and exclusion, Pigeonhole principle, generalized Pigeonhole principle and its application, permutations and combinations, circular permutations, permutations with repetitions, combinations with repetitions, permutations of sets with indistinguishable objects.

Unit4: Mathematical Logic

Connectives, truth tables, Tautologies and Contradictions, Equivalence and Implications, NAND and NOR, Normal forms- CNF, DNF, Converting expressions to CNF and DNF, Theory of inference, Propositional Calculus, Predicate calculus (only introduction), predicates and quantifiers.

Object-Oriented Programming in C++

(Core/Credit:4 (3+1))

Course Code: BCA-CORE-MC-0300304

1. Learning Outcome: After successful completion of this course, students will be able to:
   a) Will be able to imagine real-life concepts as objects; derive their properties and functions to operate.
   b) Develop programs using object- oriented features like data abstraction, polymorphism, inheritance, exception handling.
   c) Know C++ streams, operators
   d) Know file handling techniques in C++.

2. Total Credit: 4, Theory: 3, Practical: 1

3. List of reference books:
   a) M. T. Somashekara, D. S. Guru et-al; *Object-Oriented Programming with C++,* 2nd Edition, PHI,2012.
   b) Bjarne Stroustrup, *The C++ Programming Language*, Special Edition, Pearson Education, 2004.
   c) Deitel&Deitel, *C++ How to program*, Pearson Education Asia, 6th Edition, 2008
   d) Schildt Herbert, *The Complete Reference C++,* Tata McGraw Hill, 4th Edition, 2003.

4. Detailed Syllabus:

A. Theory Content

Unit 1: Introduction to object-oriented programming

Origins of C++, Basic Concepts of Object-Oriented Programming, Benefits of OOP, Applications of OOP, Introduction to C++, Structure of a Simple C++ program, Output operator, Input operator, Cascading of I/O operators, Tokens- keyword, identifiers, constants, strings and operators. Basic data types, User defined data types, Dynamic initialization of variables, Reference variables, Operators in C++, Scope resolution operator &applications, Member dereferencing operators, Memory Management operators, new and delete, Control Structures-simple if, if else, nested if, switch, while do, break and continue statements, Introduction to Functions-Function Prototyping, Call by reference, Return by reference, Inline functions, Default arguments, Constant arguments.

Unit 2: Classes and objects

Introduction - Defining a class-Class Vs structures, Creating objects, Accessing class members, Defining member functions- Outside the class definition, Inside the class definition, Outside functions as inline, Nesting of member functions, Private member functions, Memory allocation for objects, Array-Declaring an array-accessing elements of an array, Array of objects, Friendly functions, Constructors and destructors, Basic Concepts of constructors, Default constructor, Parameterized constructor, Multiple constructors in a class, Constructor with default arguments, Dynamic initialization of objects, Copy constructor, Dynamic constructors, Destructors\

Unit 3: Function and operator overloading

Overloading Concepts Function Overloading: Functions with different sets of parameters, default and constant parameters, Rules for overloading operators. Defining operator overloading, Overloading Unary operators, Prefix and Postfix operators overloading, Overloading Binary operators, overloading relational operators, Overloading using friend functions, Overloading subscript operator, Pitfalls of operator overloading, Type conversion-Basic to Class, Class to Basic

Unit 4: Inheritance

Introduction-Defining derived classes, Types of inheritances. Making a private member inheritable, multilevel inheritance, multiple inheritance, Hierarchical inheritance, Hybrid inheritance, Virtual base classes, Abstract classes, Constructors in derived classes, nesting of classes, polymorphism-Compile time and Run time polymorphism, Pointers to objects, this pointer, Pointer to derived classes, Virtual functions, Rules for virtual functions, Pure virtual functions.

Unit 5: Streams

C++ stream classes-put() and get() functions, getline() and write() functions, Overloading << and >> operators, Formatted Console I/O operations, ios class functions- width(), precision(), fill(), setf() and unsetf(), Formatting flags, Manipulators, User defined manipulators.

Unit 6: Files

Introduction-Stream classes for files, Opening files using constructor, Opening files using open(), Filemodes, Detecting end of file-eof(), Sequential input and output-put() and get()-Reading and writingobjects-read() and write()-Random Access files-Manipulating file.

B. Practical / Lab work to be performed

1. Define a class named *triangle* to represent a triangle using the lengths of the three sides. Write a constructor to initialize objects of this class, given the lengths of the sides. Also write member functions to check
   a. if a triangle is isosceles
   b. if a triangle is equilateral

   Write a main function to test your functions.
2. Define a structure *employee* with the following specifications.
   empno: integer *ename*: 20 characters *basic, hra, da* : float
   calculate() : a function to compute net pay as basic+hra+da with float return type.
   getdata() : a function to read values for empno, ename, basic, hra, da.
   dispdata() : a function to display all the data on the screen Write a main program to test the program.
3. Define a class *circle* to represent circles. Add a data member *radius* to store the radius of a circle. Write member functions area() and perimeter() to compute the area and perimeter of a circle.
4. Define a class *complex* with two data members *real* and *imag*to represent real and imaginary parts of a complex number. Write member functions

   rpart() : to return the real part of a complex number *ipart()* : to return the imaginary part of a complex number *add()* : to add two complex numbers.
   mul() : to multiply two complex numbers.
   Write constructors with zero, one and two arguments to initialize objects. (*This is an example of polymorphism.*)
5. Define a class *point* with two data members *xordinate*and*yordinate*to represent all points in the two dimensional plane by storing their x co-ordinate and y co-ordinate values. Write member functions
   dist(): to return the distance of the point from the origin.
   slope(): to return the slope of the line obtained by joining this point with the origin.
   Write constructors with zero, one and two arguments to initialize objects. Also write a friend function to compute the distance between two points.
6. Define a class *string* with the following data members char *p; int size; and write member functions to do the following (without using library function) and using dynamic memory allocation.
   a. Length of the string
   b. Compare two strings
   c. Copy one string to another
   d. Reverse the string

   Write suitable constructors and destructors. Also write a copy constructor for the class.

Information and Cyber Security

(SEC/Credit: 3)

Course Code: SEC-BCA-MC-0300103

1. Learning Outcomes: After successful completion of this course, students will be able to:
    a. Understand fundamentals of information and cyber security
    b. Identify common cyber threats and attacks
    c. Explain cryptography basics and secure communication
    d. Apply safe computing practices and cyber ethics

2. Total Credit: 3, Theory: 3, Practical: 0
3. Reference Books:
    a. Nina Godbole and Sunit Belpure, Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley.
    b. Chwan-Hwa (John) Wu and J. David Irwin, Introduction to Cyber Security, CRC Press.
    c. James Graham, Richard Howard and Ryan Olson, Cyber Security Essentials, CRC Press.
    d. Mark Stamp, Information Security: Principles and Practice, Wiley.

4. Detailed Syllabus:

A. Theory:

Unit 1: Introduction to Information & Cyber Security

Information Security – Definition and importance, Cyber Security – Scope and need, Difference between Information Security and Cyber Security, CIA Triad – Confidentiality, Integrity and Availability, Assets, Threats, Vulnerabilities and Risks, Types of attackers, Layers of security.

Unit 2: Cyber Threats and Attacks

Types of cyber attacks – Active and Passive attacks, Insider and Outsider attacks, Malware – Virus, Worm, Trojan Horse, Spyware, Adware, Ransomware, Phishing, Spoofing, Sniffing, Man-in-the-Middle attacks, DoS and DDoS attacks, Social engineering attacks.

Unit 3: Cryptography and Data Security

Introduction to cryptography, Need for cryptography, Plain text and Cipher text, Encryption and Decryption, Symmetric key cryptography – DES, AES (conceptual), Asymmetric key cryptography – Public and Private keys, RSA algorithm (basic idea), Hash functions – MD5, SHA, Digital Signature, Digital Certificates, Applications of cryptography in real life.

Unit 4: Network and System Security

Network security basics, Firewalls – Types and functions, Antivirus and Anti-malware tools, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), Authentication techniques, Access control, Operating system security basics, Mobile and Wi-Fi security.

Unit 5: Cyber Laws and Ethics

Cyber laws and cyber crimes, Information Technology Act 2000 – overview, Cyber ethics, Privacy and data protection, Cyber safety practices, Social media security issues, Career opportunities in cyber security.

3rd Semester

Basic of Operating System and Networking

(For Minor Students- BA, BSc / Credit: 4(3+1))

Course Code: BCA-MN-MC-0300404

1. Learning Outcome:
   - To give students the role of operating system in computer.
   - To provide students the concept of process management
   - To familiarize students with the concept of Networking
   - To provide the practical concept of OS and Networking..
2. Course Type: Compulsory
3. Total Credit: 4, Theory: 3, Practical: 1
4. Number of required hours:
   a) Theory: 45 hrs
   b) Practical: 30 hrs

5. List of reference books

(a) Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Wiley

(b) Data Communications and Networking, Behrouz A. Forouzan, Sophia Chung Fegan, McGraw-Hill,

(c) Operating Systems: Internals and Design Principles, William Stallings, Pearson

(d) Computer Networks, Andrew Stuart Tanenbaum, Pearson Education

6. Detailed Syllabus

Unit No.	Unit content	No.of classes
1	Introduction to Operating System Types of software, application software, system software, operating system, types of operating systems, operating system structure, system calls, protection and security, history and generations of operating system, Open source operating systems	10
2	Process and Memory Management Process, process states, process scheduling algorithms, scheduling criteria, turn around time, waiting time, response time, First Come First Serve (FCFS) scheduling, Shortest-Job-First (SJF) Scheduling, thread, deadlock, virtual memory	13

3	Networking Introduction Computer network, network topologies, LAN and WAN, internet, network models: ISO OSI, TCP/IP, protocols and standards, transmission media: guided and unguided transmission media, twisted pair cable, co-axial cable, fiber optic cable, radio waves, micro waves, infrared	10
4	Network Concepts and Internet Analog and digital signal, error detection and correction, Ipv4 addressing, Ipv6 addressing Internet, ISP, types of ISPs, WWW, web server, web client, HTTP, HTTPS, web browsers, domain name, URL, DNS, hierarchy of name servers, electronic mail, SMTP, POP, IMAP, FTP	12

Practical Part

1. Installation of Windows operating system
2. Commands in windows: cd, mkdir, ren, assoc, attrib, mv, cp, cls, del, dir, rmdir etc.
3. Installation of Linux operating system
4. Commands in linux: ls, mkdir, cd, touch, pwd, chmod, cat, echo, su, rm, mv, cp, locate, ps, top, man etc.
5. Installing open source software in linux
6. Networking commands and tools in windows: ipconfig, nslookup, ping, netstat, arp, tracert, getmac etc.
7. Networking commands and tools in linux: ifconfig, ip, traceroute, tracepath, ping, netstat, ss, dig, nslookup, arp, hostname, whois etc.
8. Draft a formal or semi-formal Email:
   a) To a university's admission office to inquire about application process
   b) To your boss to ask for a day off
   c) To a hiring manager after an interview for a software design position