



University of Engineering and Management
Institute of Engineering & Management, Salt Lake Campus
Institute of Engineering & Management, New Town Campus
University of Engineering & Management, Jaipur



2nd Semester Syllabus for BCA Admission Batch 2024

Index:

Content	Page No.
Syllabus Structure	3
Data Structures(BCACC201)	4-13
Object oriented programming(BCACC202)	14-19
Computer Architecture(BCACC203)	20-25
Mathematics for Computing(BCAMD201)	26-28

BCA 2024 Course structure									
SEMESTER 2									
SL NO	TYPES OF COURSE	SUB CODE	SUB NAME	L	T	P	S	TOTAL CONTACT HRS	CREDIT POINTS
THEORY									
1	Computer Science & Application	BCACC201	Data Structures	3	1	0	0	4	4
2	Computer Science & Application	BCACC202	Object oriented programming	3	1	0	0	4	4
3	Computer Science & Application	BCACC203	Computer Architecture	3	1	0	0	4	4
4	Multidisciplinary	BCAMD201	Mathematics for Computing	3	1	0	0	4	2
5	Value Added Course	BCAESP201	General Studies & Current Affairs -II	2	0	0	0	2	2
PRACTICAL									
6	Computer Science & Application	BCACC291	Data Structure lab with C	0	0	2	0	4	2
7	Computer Science & Application	BCACC292	Object oriented programming lab with C++	0	0	2	0	4	2
SESSIONAL									
8	Skill Enhancement	BCASDP281	Competitive Aptitude Training –II	0	0	0	2	2	2
9	Ability Enhancement	BCAAE201	Principles of Management	0	0	0	2	2	2
MOOCS/MAR/IFC									
10	Value Added Course	MOOCs	Massive Open Online Course	0	0	0	0	0	0
11	Value Added Course	IFC	Industry and foreign certification	0	0	0	0	0	0
12	Value Added Course	MAR	Mandatory Additional Requirements(MAR)	0	0	0	0	0	0
								30	24



University of Engineering and Management
Institute of Engineering & Management, Salt Lake Campus
Institute of Engineering & Management, New Town Campus
University of Engineering & Management, Jaipur



Syllabus for BCA Admission Batch 2024

Subject Name: **Data Structures**

Credit: 4

Lecture Hours: 40

Subject Code: **BCACC201**

Pre-requisite: Fundamental computer knowledge

Relevant Links

[Study Material](#)

[Coursera](#)

[LinkedIn Learning](#)

[NPTEL](#)

COURSE OBJECTIVES:

1. Allow to assess how the choice of data structures and algorithm design methods impacts the performance of programs
2. To choose the appropriate data structure and algorithm design method for a specified application.
3. To solve problems using data structures such as linear lists, stacks, queues, binary trees and binary search trees writing programs for these solutions.
4. To efficiently implement the different data structures and solutions for specific problems.

COURSE OUTCOMES:

CO1: Analyze algorithms for time and computational complexity.

CO2: Select appropriate search techniques (Linear Search, Binary Search, Hashing) based on the problem's nature.

CO3: Implement and apply Stacks, Queues, and Linked Lists to various problems.

CO4: Use non-linear data structures for search, insertion, and retrieval; evaluate time complexity of balanced and unbalanced trees, and apply these to relevant problems.

Module No	Topic	Sub Topic	Chapter Name	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignment
1	Introduction	Overview of fundamental concepts	An Introduction to Data Structures. Chapter 1, Data Structures Through C (A practical Approach) G.S.BALUJA	International Academia: https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-1-algorithms-and-computation/ Industry Mapping: C Lion, Eclipse(IDE), Visual Studio, CLion, Clang	6	1. Write a c program to print even numbers using a loop. 2. Write a c program to print the sum of n natural numbers using a loop. 3. Create a student database using structure. 4. Write a C program that allows the user to input details of multiple students (name, roll number, and marks). Implement a function to search for a student by roll number and display their details if found. 5. Write a C program to generate the Fibonacci series using recursion. Ask the user to enter the number of terms in the series and then display the Fibonacci series up to that number of terms.
	Data Structure	Definition of data structure, Data structure operation	An Introduction to Data Structures Data Structures Through C (A practical Approach) G.S.BALUJA . Chapter			

			1,(1.10)			
	Algorithms	Complexity of Algorithms, Time Space trade off, Asymptotic Notations for Complexity of Algorithms, Subalgorithms , Role of variables in data structures and algorithms, Importance and types of data in computing Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, Multi-	Data Structures Through C (A practical Approach) G.S.BALUJA An Introduction to Data Structures . Chapter 1(1.6- Algorithms)			

		dimensional arrays, Matrices, Sparse Matrices				
2	Arrays		Data Structures Through C (A practical Approach) G.S.BALUJA Arrays Chapter 3	International Academia: https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-2-data-structures-and-dynamic-arrays/ Industry Mapping: C Lion, Eclipse (IDE), Visual Studio, CLion, Clang	14	1. Write a C program to input elements into an array of integers and calculate their sum and average. Take the input from user to enter the size of the array and its elements. Display the sum and average of the elements. 2. Develop a C program to reverse the elements of an array of integers. Ask the user to input the size of the array and its elements. Reverse the elements of the array in-place and then display the reversed array. 3. Create a C program to sort an array of integers in ascending order. Allow the user to input the size of the array and its elements. Implement any sorting algorithm (e.g., bubble sort, selection sort) to sort the elements, and then display the sorted array. 6. Write a C program to create a linked list of integers. Implement functions to insert elements at the beginning, end, and at a specified position in the linked list. Also, include a function to display the elements of the linked list.

	Linked Lists	Linked Lists, Representation in memory, Traversing, Searching, Memory allocation (Garbage collection, overflow, underflow), Insertion, Deletion, Circular linked lists, Doubly linked lists, Header linked lists	Data Structures Through C (A practical Approach) G.S.BALUJA Linked List Chapter 7			7. Develop a C program to reverse a linked list. Implement a function to reverse the elements of a given linked list in-place and then display the reversed linked list. 8. Develop a C program to reverse a linked list. Implement a function to reverse the elements of a given linked list in-place and then display the reversed linked list.
	Stacks	Definition, Array representation, Linked representation, Polish notation, Evaluation of Postfix Expressions, Transforming Infix to Postfix Expressions	Data Structures Through C (A practical Approach) G.S.BALUJA Stacks Chapter 5			1. Write a C program to check whether the given expression has balanced parentheses using a stack. The program should ask the user to input an expression containing only parentheses (such as "({})" or "{[]}" etc.). Implement a stack data structure and use it to verify if the parentheses are balanced or not. 2. Develop a C program to evaluate a postfix expression using a stack. Take input from user a postfix expression (e.g., "53+2*") and implement a stack-based algorithm to evaluate the expression and display the result.

	Queues	Definition, Array representation, Linked representation, Circular queues, Priority Queue, D-Queue	Data Structures Through C (A practical Approach) G.S.BALUJA Queues Chapter 6			1. Write a C program to implement a queue data structure. Implement functions to enqueue (insert) elements into the queue, dequeue (remove) elements from the queue, and display the elements of the queue. 2. Develop a C program to implement a circular queue data structure. Implement functions to enqueue, dequeue, and display elements in a circular queue. 3. Design a C program to implement a queue using a linked list. Implement functions to enqueue, dequeue, and display elements in the linked list-based queue. 4. Create a C program to implement a priority queue using arrays. Implement functions to enqueue elements with priority, dequeue elements based on their priority, and display the elements of the priority queue. 5. Write a C program to simulate the functioning of a ticket counter using a queue. Implement functions to simulate customers arriving at the ticket counter and getting served. Display the sequence of customers served.
--	--------	---	---	--	--	--

3	Trees	Introduction and Definition of Trees, Tree Terminology	Trees C+ hapter 8	International Academia: 1. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-6-binary-trees-part-1/ 2. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-7-binary-trees-part-2-avl/ IndustryMapping:C Lion, Eclipse(IDE), Visual Studio, CLion, Clang	10	1. Write a C program to create a binary search tree (BST) of integers. Implement functions to insert elements into the BST, perform inorder, preorder, and postorder traversals, and display the elements accordingly. 2. Develop a C program to calculate the height of a binary tree. Implement a function to calculate the height (depth) of the binary tree and display the result. 3. Create a C program to find the mirror image of a binary tree. Implement a function to convert the given binary tree into its mirror image and display the mirror image tree. 4. Design a C program to perform operations on a binary search tree (BST). Implement functions to search for an element in the BST, delete an element from the BST, and display the resulting BST after each operation. 5. Write a C program to perform level order traversal of a binary tree. Implement a function to traverse the binary tree level by level and display the elements in each level.
---	-------	--	-------------------	--	----	---

	Binary Tree	Representing Binary Trees in Memory, Traversing Binary Tree: Preorder traversal, In-order traversal, Post-order traversal, Traversal algorithms using stacks	Trees Chapter 8			
	Binary Search Trees	Searching in Binary Search Trees, Inserting in Binary Search Trees, Deleting in a Binary Search Tree	Trees Chapter 8			
	Advanced Tree Structures	AVL trees and implementations, M-trees, B-Trees (definition only)	Data Structures Through C (A practical Approach) G.S.BALUJA Trees Chapter 8			
4	Searching	Sequential Search, Binary Search, Indexed	Searching and Hashing Chapter 8	International Academia: https://ocw.mit.edu/courses/6-006-introduction-to-	10	1. Write a C program to perform linear search on an array of integers. Implement a function to search for a given element in the array using linear search. Ask the

		Search		algorithms-spring-2020/resources/lecture-3-sets-and-sorting/ https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-4-hashing/ https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-5-linear-sorting/	user to input the size of the array and its elements, as well as the element to be searched. Display the index of the element if found, otherwise indicate that the element is not present in the array. 2. Develop a C program to perform binary search on a sorted array of integers. Implement a function to search for a given element in the sorted array using binary search. Ask the user to input the size of the array and its elements in ascending order. Then, ask for the element to be searched and use the binary search algorithm to find its index. Display the index of the element if found, otherwise indicate that the element is not present in the array.
	Sorting	Introduction and Notation, Insertion Sort, Selection Sort, Shell Sort, Divide And Conquer, Merge sort for Linked List, Quick sort for Contiguous List.	Sorting Chapter 10		3. Develop a C program to sort an array of integers in ascending order using the merge sort algorithm. Implement a function to perform merge sort on the array. Ask the user to input the size of the array and its elements. After sorting, display the sorted array. 4. Write a C program to sort an array of integers in ascending order using the bubble sort algorithm. Implement a function

						to perform bubble sort on the array. Ask the user to input the size of the array and its elements. After sorting, display the sorted array.
	Hashing	Sparse Tables, Choosing a Hash function, Collision Resolution with Open Addressing, Collision Resolution by Chaining.	Data Structures Through C (A practical Approach) G.S.BALUJA Searching and Hashing Chapter 8			Write a C program to implement a hash table data structure. Implement functions to insert key-value pairs into the hash table, retrieve the value associated with a given key, and handle collisions using chaining. Ask the user to input keys and values, and demonstrate the functionality of the hash table

TEXT BOOK:

1. Seymour Lipschutz, “Data Structures with C”, Schaum’s out Lines, Tata Mc Graw Hill, 2011.
2. Data Structures Through C (A practical Approach) G.S.BALUJA
3. Robert Kruse, C.L.Tondo, Bruce Leung, Shashi Mogalla, “Data Structures and Program Design using C”, Pearson Education, 2009.

REFERENCE BOOKS:

1. Mark Allen Weiss, “ Data Structures and Algorithm Analysis in C”, Second Edition, Pearson Education, 2013
2. Reema Thareja, “Data Structures Using C”, Oxford Universities Press 2014, 2nd Edition.
3. Alfred V Aho, John E Hopcroft and Jeffery D Ullman, “ Data Structures and Algorithms”, Pearson Education.
4. Samiran Chattopadhyay, Debabrata Ghosh Dastidar and Matagini Chattopadhyay, “ Data Structures through C Language”, BPB Publication.

List of equipment/apparatus for laboratory experiments:

Sl. NO	Requirements
1.	Computer with moderate configuration
2.	A programming language compiler



University of Engineering and Management
Institute of Engineering & Management, Salt Lake Campus Institute of Engineering
& Management, New Town Campus University of Engineering & Management, Jaipur
Syllabus for BCA Admission Batch 2024

Subject Name: **Object Oriented Programming**

Credit: 4

Lecture Hours: 40

Subject Code: **BCACC202**

Pre-requisite: Basics knowledge of programming language, Logic building skills.

Relevant Links:

[Study Materials](#)

[Coursera](#)

[LinkedIn Learning](#)

[Infosys Springboard](#)

[NPTEL](#)

COURSE OBJECTIVES:

1. To introduce students to a powerful programming language
2. To understand the basic structure of object oriented program
3. To gain knowledge of various programming errors
4. To enable the students to make flowchart and design an algorithm for a given problem
5. To enable the students to develop logics and programs

COURSE OUTCOMES:

CO 1: Students will be able to learn different programming techniques using object-oriented technology with C++.

CO 2 Students will be able to learn how to solve real life problems by implementing data, security, reuse of code, polymorphism etc.

CO 3: Students will be able to learn how to solve real life problems by using pointer, virtual function and FILE handling.

CO 4: Students will be able to learn how to solve real life problems by implementing exception handling and generic programming.

Module Number	Topic	Sub topics	Chapter No. (Text Book)	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignment
1	Introduction of Object Oriented Programming, Introduction of C++ Tokens and Keywords, Function	A look of object oriented programming, Object oriented programming paradigm, Basics of object oriented programming, Benefits of object oriented programming, Object oriented languages, Applications of object oriented programming, Application of C++, A simple C++ program, More C++ statements, An example with Class, Structure of C++ program, Creating a source file, Compiling and linking Tokens, Keywords, Identifiers and constants, 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, Memory management operators, Manipulators, Type cast operator, Expressions and their types, Special assignments expressions, Implicit	1, 2, 3, 4	International Academia: https://ocw.mit.edu/course/s/6-096-introduction-to-c-january-iap-2011/ Industry Mapping: CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)	8	1. Console based C++ program to display, insert, search and sort elements of an array, prime no, factorial, Fibonacci series, palindrome etc.

		conversions, Operator overloading, Operator precedence, Control structures Introduction, The main function, Function prototyping, Call by reference, Return by reference, Inline functions, Default arguments, Const. arguments, Function overloading, Friend and virtual functions, Math library function				
2	Introduction to class, Constructor and Destructor, Operator Overloading, Inheritance	Introduction, Specifying a class, Defining a member functions, A C++ program with class, Making an outside function inline, Nesting a member functions, Private member function, Arrays within a class, Memory allocation for objects, Static data members, Static member functions, Arrays of objects, Objects as function arguments, Friendly function, Returning objects, Const member functions, Pointers to member, Local classes. Constructors, Parameterized Constructors, Multiple Constructors in a class, Constructors with default arguments, Dynamic initialization of objects, Copy Constructors, Dynamic	5, 6, 7, 8	International Academia: https://ocw.mit.edu/course/s/6-096-introduction-to-c-january-iap-2011/ International Academia: https://ocw.mit.edu/course/s/6-s096-introduction-to-c-and-c-january-iap-2013/pages/lectures-and-assignments/c-inheritance/ Industry Mapping: CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)	12	1. Program based on object, class, and encapsulation property 2. Program based on array of object 3. Program based on various (default, parameterize, copy) constructor 4. Constructor overloading program 5. Destructors program 6. Program based on various operator overloading 7. Program based on function overloading 8. Program based on friend function 9. Program based on various type of inheritance

		Constructors, Constructing two dimensional array, Const objects, Destructors, Defining operator overloading, Overloading unary operators, Overloading binary operators, Overloading unary operators using friends, Manipulation of string using operators, Rules for Overloading operators, Type conversions Defining derived classes, Single inheritance, Making private member inheritable, Multilevel inheritance, Multiple inheritance, Hierarchical inheritance, Hybrid inheritance, Virtual base classes, Abstract classes, Constructors in derived classes, Member classes: nesting of classes				
3	Pointer and virtual function C++ Streams FILE handling in C++	Introduction, Pointers, Pointers to objects, this pointer, Pointers to derived classes, Virtual functions, Pure Virtual functions C++ streams, C++ stream classes, Unformatted I/O operators, Formatted I/O operations, Managing output with manipulator	9, 10, 11	International Academia: https://ocw.mit.edu/course/s/6-096-introduction-to-c-january-iap-2011/ Industry Mapping: <i>CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)</i>	10	1. Program based on pointer concept 2. Virtual and Pure virtual function programs 3. File and streams based programs

		Classes for File stream operations, Opening and closing a File, Detecting end-of-file, More about Open();File modes, File pointers and their manipulations, Sequential input and output operations, Updating a File: Random access, Error handling during File operations, Command-line arguments.				
4	Template, Exception Handling, Standard Template Library	Introduction to generic programming, Class templates, Class templates with multiple parameters, Function templates, Function templates with multiple parameters, Overloading of template functions, Member function templates, Non-type template arguments Introduction of exception, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions Components of STL, Containers, Algorithms, Iterators, Application of	12, 13	International Academia: https://ocw.mit.edu/course/s/6-096-introduction-to-c-january-iap-2011/ Industry Mapping: CLion, Eclipse (IDE), Sublime Text, Atom (Text Editors), Visual Studio Debugger (Debuggers)	10	1. Function and class template based program 2. Program based on various type of exceptions.

		container classes, Function objects				
--	--	-------------------------------------	--	--	--	--

Learning Resources:

Text Books:				
Name of Author	Title of the Book	Edition/ISSN/ISBN	Name of the Publisher	
E. Balagurusamy	Object Oriented Programming with C++	Fourth Edition	Tata McGraw Hill	Download eBook
Reference Books:				
Herbert Schildt	C++: The Complete Reference	Fourth Edition	Tata McGraw Hill	Download eBook
List of equipment/apparatus for laboratory experiments:				
SL. No.				
1	Computer with moderate configuration			
2	g++ compiler and other software's as required			
3	C++ compiler (online)	Online compiler-1	Online compiler-2	



University of Engineering and Management
Institute of Engineering & Management, Salt Lake Campus
Institute of Engineering & Management, New Town Campus
University of Engineering & Management, Jaipur



Syllabus for BCA Admission Batch 2024

Subject Name: **Computer Architecture**

Credit: 4

Lecture Hours: 40

Subject Code: **BCACC203**

Pre-requisite: **Digital Electronics**

Relevant Links:

[Study Material](#)

[Coursera](#)

[LinkedIn Learning](#)

[NPTEL](#)

Course Objectives:

1. Basic revision of Boolean algebra and their simplification.
2. To study the Register Transfer Language and Microoperations
3. To introduce the computer registers timing and control, instruction and interrupt cycle.
4. To illustrate the concept Central Processing Unit, Input/Output operations and Memory Management.

Course Outcomes:

- CO1 Upon completion of this course, student will be able to understand basic structure of computer and perform computer arithmetic operations
- CO2 Ability to understand control unit operations with design memory organization that uses banks for different word size operations
- CO3 Understand the concept of register transfer language and basic microoperations
- CO4 Understand the concept of I/O organization and conceptualize instruction level parallelism.

Module number	Topic	Chapter Name	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignment
1	Data Representation	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 4	Data type, Complement, Fixed point Representation, Floating point representation, other binary code.	International Academia: (https://ocw.mit.edu/courses/6-823-computer-system-architecture-fall-2005/resources/l02_fifties) Industry Mapping: Software: Xilinx Vivado, Icarus Verilog, Hardware: FPGA Kit, Basys3 board	8	1. Execute the HDL programs to draw the circuits for basic gates. 2. Develop an HDL code snippet that performs a combination of arithmetic and logic micro-operations on two binary numbers.
	Register Transfer and Micro operations.	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 5	Register Transfer Language, Register Transfer, Bus and Memory Transfers, Register Transfer Micro-operations, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations			
2	Basic Computer Organization and Design	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 6	Instruction code, computer register, computer Instruction, timing and control, Instruction cycle, Memory reference instruction, Input-output and interrupt, complete computer description, Design of basic computer, and design of accumulator logic	International Standards :(https://ocw.mit.edu/courses/6-823-computer-system-architecture-fall-2005/resources/l04_micropr) Industry Mapping: Software: Xilinx Vivado, Icarus Verilog, Hardware: FPGA Kit, Basys3 board	12	1. Design an HDL code that defines an instruction code for a simple operation (e.g., addition) and implements the corresponding operation using computer registers. 2. Write an HDL code to implement a 2-bit

	Computer Arithmetic	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 11	Introduction, Addition and subtraction, Multiplication algorithm, division algorithm, Floating point arithmetic operation, Decimal arithmetic unit, Decimal arithmetic operation			full adder at the behavioral level.
	Programming the Basic Computer	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 7	Introduction, Machine Language, Assembly Language, The assembler, Programming loop, Programming arithmetic and logic operation, subroutine, Input Output programming			3. Create an HDL implementation for a basic computer, including the ALU, registers, and control unit components.
3	Memory Organization	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 13	Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware	International Standards : (https://ocw.mit.edu/courses/6-823-computer-system-architecture-fall-2005/resources/107_caches/) Industry Mapping: Software: Xilinx Vivado, Icarus Verilog, Hardware: FPGA Kit, Basys3 board	10	1. Design a circuit for implementing any arithmetic instruction in a pipeline. Also, develop the instruction and interrupt pipeline concept. Justify the circuit with a proper simulation 2. Let A0, A1, A2, and

	Central Progressing Unit (CPU)	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 9	Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC)		A3 be the four inputs, S0 and S1 be the select line inputs, and Y be the output for illustrating a multiplexer. Use Dataflow architecture for drawing the circuit. Show the schematic diagram, Truth table, and Karnaugh Map. 3. Use the following equation $\text{Sum} = x \oplus y \oplus \text{Cin}$, $\text{Carry} = (x \text{ AND } y) \text{ OR } (\text{Cin AND } (x \oplus y))$ to develop a Full adder using two Half adders using the concept of components and port map. Show the schematic diagram, Truth table, and Karnaugh Map. Write an HDL code to implement a D-type latch at the behavioral level.
--	---	---	--	--	--

4	Pipeline and Vector Processing	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 10	Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors	International Standards: (https://ocw.mit.edu/courses/6-823-computer-system-architecture-fall-2005/resources/106_pipeline/) Industry Mapping: Software: Xilinx Vivado, Icarus Verilog, Hardware: FPGA Kit, Basys3 board	10	1. Implement an 8-bit right shift register with a parallel load facility. 2. Write an HDL code to implement a 32-bit Booth's multiplier using behavioral style. Write an HDL code using switch-level modeling to implement the function $F = (A'.B + C.D)'$ using CMOS logic. You may use the nmos and pmos transistors for the purpose.
	Input-Output Organization	Computer System Architecture, M.Morris Mano, PEARSON, Chapter 12	Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA), Input-Output Processor (IOP), Serial Communication			

TEXT BOOK:

1. Computer System Architecture, M.Morris Mano, PEARSON

REFERENCE BOOKS:

1. Computer Organization & Architecture-Designing for performance, William Stallings, PERSON
2. Computer Architecture & Organization, J.P.Hayes, Tata McGraw Hill

List of equipment/apparatus for laboratory experiments:

Sl. NO	Requirements
1.	Computer with moderate configuration
2.	Xilinx Software



University of Engineering and Management
Institute of Engineering & Management, Salt Lake Campus
Institute of Engineering & Management, New Town Campus
University of Engineering & Management, Jaipur



Syllabus for BCA Admission Batch 2024

Subject Name: **Mathematics for Computing**

Credit: 2

Lecture Hours: 40

Subject Code: **BCAMD201**

Pre-requisite: Basic knowledge in number system and geometry

Relevant Links:

[Study Material](#)

[Coursera](#)

[Linkedin](#)

[NPTEL](#)

COURSE OBJECTIVES:

1. To understand and apply Probability concepts
2. To analyze data using descriptive statistics
3. To apply inferential statistics for decision-making
4. To develop problem-solving and critical thinking skills

COURSE OUTCOMES:

CO1 Students will be able to understand and apply fundamental concepts of probability, including definitions, rules, and theorems, to analyze and interpret real-world situations involving uncertainty.

CO2 Students will be able to develop proficiency in using descriptive and inferential statistical methods to summarize data, make predictions, and draw conclusions based on sample data, including the application of confidence intervals and hypothesis testing.

CO3 Students will be able to gain skills in collecting, organizing, and analyzing data using appropriate statistical techniques and tools, enabling students to effectively communicate results and insights derived from their analyses.

CO4 Students will be able to develop the ability to create and interpret various statistical charts and graphs (such as histograms, pie charts, box plots, and scatter plots) to effectively visualize data trends, patterns, and distributions, enabling informed decision-making based on statistical analysis.

Module No	Topic	Sub Topic	Chapter Name	Mapping with Industry and International Academia	Lecture Hours
MODULE 1	Basics of Probability Theory	Introduction, Introductory Definitions, Classical Definition of Probability, Theorems on Probability & Addition Theorem, Axiomatic definition of Probability, Conditional Probability, Multiplication Rule of Probability, Independent Events, Baye's Theorem.	Basics of Probability Theory, Chapter-3.1, BCA Mathematics Vol-III	<i>International Academia:</i> Introduction to Probability and Statistics <i>Industry Mapping:</i> Matlab	10
MODULE 2	Collection of Data and Frequency Distribution Charts and Diagram	Introduction, Primary Data, Secondary Data, Method of Collection of Primary Data, Source of Secondary Data, Classification and Tabulation of Data, Tabulation of Data and Statistical table. Statistics and its Related Terms, Frequency Distribution, Diagram, Bar Chart, Pie-Chart, Line Chart, Histogram of a Frequency Distribution, Frequency Polygon, Cumulative Frequency Polygon or Ogive	Collection of Data Chapter-4.1, BCA Mathematics Vol-III Frequency Distribution Charts and Diagram, Chapter-4.2, BCA Mathematics Vol-III	<i>International Academia:</i> Introduction to Probability and Statistics <i>Industry Mapping:</i> Matlab	10
MODULE 3	Measures of Central Tendency	Introduction, Mean, Arithmetic Mean(AM), Geometric Mean(GM), Harmonic Mean(HM), Median, Mode, Quartiles, Calculation of	Measures of Central Tendency,	<i>International Academia:</i> Introduction to Probability and Statistics	10

		Quartiles, Deciles, Calculation of Deciles, Percentiles, Calculation of Percentiles.	Chapter-4.3,BCA Mathematics Vol-III	<i>IndustryMapping:</i> <i>Matlab</i>	
MODULE 4	Measures of Dispersion	Introduction, Classification of Measure of Dispersion, Range, Quartile deviation(Semi inter quartile range), Mean deviation(Mean absolute deviation), Variance and Standard Deviation, Relative Measures of Dispersion	Measures of Dispersion, Chapter-4.4,BCA Mathematics Vol-III	<i>International Academia:</i> <u>Introduction to Probability and Statistics</u> <i>IndustryMapping: Matlab</i>	10

TEXT BOOK:

BCA MATHEMATICS, Vol-III, B.K.PAL, K.DAS,Fourth Edition

REFERENCE BOOKS:

1. Fundamental Treatise on Probability and Statistics, Dr. Arup Mukherjee