Detailed Syllabus
Lecture-wise Breakup

Subject Code **NBA Code**	23B61CA211 CBAC201	**Semester: Odd** **(specify Odd/Even)**	**Semester** III **Month from: July to December 2024**
Subject Name	Algorithm and Problem Solving		
Credits	**3-0-0**	**Contact Hours**	**3**

Faculty (Names)	**Coordinator(s)**	Dr. Meenal Jain
	Teacher(s) (Alphabetically)	Dr. Meenal, Dr. Shelendra Pal

COURSE OUTCOMES		**COGNITIVE LEVELS**
CO1	Describe different types of problems and basic concepts of algorithm design.	Understand Level (Level 2)
CO2	Apply various data structure operations and algorithm strategies to solve a given problem.	Apply Level (Level 3)
CO3	Analyze and identify an appropriate data structure and/or design strategy to develop an efficient algorithm for a given problem.	Analyze Level (Level 4)
CO4	Evaluate a given problem and develop an efficient solution to a given problem.	Evaluate Level (Level 5)

Module No.	**Subtitle of the Module**	**Topics in the Module**	**No. of Lectures for the module**
1.	Introduction	Introduction to problem solving approach; Asymptotic Analysis: Growth of Functions and Solving Recurrences; Notations- Big O, big omega, big theta; Empirical analysis of sorting and searching algorithms – Merge sort, Quick sort, Binary search (aligned with Module 3 in detail).	3
2.	Search Trees and Priority Queue	Search Tree, Binary Heaps and Min-Max Heaps	3
3.	Design Technique: Divide and Conquer	Fundamentals of Divide and Conquer (D&C) approach using Binary search, Quick sort, and Merge sort;	4
4.	Design Technique: Greedy Algorithms	Introduction to greedy based solution approach; Minimum Spanning Trees (Prim's and Kruskal algorithms); Shortest Path using Dijkstra's algorithm; Fractional and 0/1 Knapsack; Coinage problem	7
5.	Design Technique: Backtracking Algorithms	Review of backtracking based solution approach using N queen, and Rat in a maze; Travelling salesman problem;	5
6.	Dynamic Programming	Fundamentals of Dynamic programming-based solution approach; 0/1 Knapsack; Shortest path using Floyd Warshall;	7

7.	Dynamic Programming	Coinage problem; Longest common subsequence; Longest increasing sequence etc.	7
8.	Problem solving by search	Uninformed search (BFS, DFS)	4
9.	Tractable and Non-Tractable Problems	Efficiency and Tractability, P, NP, NP-Complete, NP- Hard problems	2
		Total number of Lectures	**42**

Evaluation Criteria

T1	20 Marks
T2	20 Marks
End Semester Examination	35 Marks
TA	25 Marks(Attendance(05)/Assignment(10)/PBL(10)))

Project based learning: Each student in a group of 3-4 will have to develop a mini project based on data structures algorithms. The students can opt any real-world application where these algorithms can be applied. The students have to implement the mini project using C/C++/Java language. Project development and its presentation will enhance coding skills, knowledge and employability of the students in IT sector.

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. **(Reference Books,** Journals, Reports, Websites etc. in the IEEE format)

1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, MIT Press, 3rd Edition, 2009
2.	Steven Skiena,The Algorithm Design Manual, Springer; 2nd edition, 2008
3.	Knuth, The art of Computer Programming Volume 1, Fundamental Algorithms, Addison-Wesley Professional; 3rd edition,1997
4.	Horowitz and Sahni, Fundamentals of Computer Algorithms, Computer Science Press, 2008
5.	Sedgewick, Algorithms in C, 3rd edition. Addison Wesley, 2002
6.	Alfred V. Aho, J.E. Hopcroft, Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley Series in Computer Science and Information Processing, 1983
7.	ACM Transactions on Algorithms (TALG)
8.	Algorithmica Journal, Springer
9.	Graphs and Combinatorics, Journal, Springer
10.	The ACM Journal of Experimental Algorithmics
11.	https://online.stanford.edu/courses/soe-ycsalgorithms1-algorithms-design-and-analysis-part-1 https://online.stanford.edu/courses/soe-ycs0001-algorithms-design-and-analysis-part-2 https://in.coursera.org/specializations/algorithms

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. **(Text books)**

1.	T. Roughgarden, *Algorithms Illuminated: Part 1: The Basics*, 1st ed., Soundlikeyourself Publishing, Sept. 2017.
2.	.T. Roughgarden, *Algorithms Illuminated: Part 2: Graph Algorithms and Data Structures*, 1st ed., Soundlikeyourself Publishing, 2018.
3.	T. Roughgarden, *Algorithms Illuminated: Part 3: Greedy Algorithms and Dynamic Programming*, 1st ed., Soundlikeyourself Publishing, 2019.
4.	M. A. Weiss, *Data Structures and Algorithm Analysis in C++*, 4th ed., Pearson, 2014.

CO Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2
CO1	2	1	1			2	1	1	2
	Moderately related to understanding the logic behind algorithm	Slightly related to understanding the concepts of algorithms	Slightly Mapped as understanding of fundamentals of algorithms for given problem.			Moderately related to understanding the concepts of algorithms	Slightly mapped as algorithms concepts learned are always used.	slightly Mapped as understanding of fundamentals of algorithms to design solutions	Moderately Mapped as understanding of fundamentals of algorithms to develop programming proficiency
CO2	3	2	1			2	1	1	2
	Strongly related to apply data structure operations and algorithm	Moderately mapped to the analysis of algorithms.	Slightly mapped to complex problem investigation.			Moderately mapped as to the analysis of algorithms.	Slightly mapped as analysis of algorithm is used throughout their life as software developers.	slightly Mapped algorithm strategies to design solutions	Moderately Mapped as algorithm strategies to develop programming proficiency
CO3	1	2	3			2	1	1	3
	Slightly mapped to analysis of designed algorithm	Moderately mapped to analysis of designed algorithm	Strongly mapped to development of solution.			Moderately mapped to usage of design techniques individually and as a team.	Slightly mapped to use of design techniques which will help them life- long	slightly mapped to investigations of designed solution of given problem	Strongly mapped as algorithm strategies to develop programming proficiency

CO4	1	2	2			3	2	1	2
	Slightly mapped to evaluation of problem	Moderately mapped to evaluation of problem	Moderately mapped to designing and analysis of efficient solution.			Strongly mapped to development of solution as a team	Moderately mapped to lifelong learning	Slightly Mapped as design and analysis of algorithm is used to for real-life applications.	Moderately mapped as algorithm strategies to develop programming proficiency
NBA Code:	**1.75**	**1.75**	**1.75**			**2.25**	**1.25**	**1.00**	**2.25**

Probability and Statistics (23B31MA211)

Course Description

Course Code	23B31MA211	**Semester: Odd**	**Semester: III, Session:** 2024-25 **Month:** July to Dec 2024
Course Name	Probability and Statistics		
Credits	3	**Contact Hours**	3-0-0
Faculty (Names)	**Coordinator(s)**	Dr. Gaurav Aggarwal	
	Teacher(s) (Alphabetically)	Dr. Ankit Kumar, Dr. Gaurav Aggarwal	

COURSE OUTCOMES		**COGNITIVE LEVELS**
After pursuing the above -mentioned course, the students will be able to:		
CO1	recall the graphical representation of data and measures of central tendency.	Remembering (C1)
CO2	explain the concepts of permutation and combination, statistics and probability theory.	Understanding (C2)
CO3	make use of probability distributions, hypothesis testing, curve fitting, correlation and regression in solving related problems.	Applying (C3)
CO4	analyze data by applying appropriate statistical techniques.	Analyzing (C4)

Module No.	**Title of the Module**	**Topics in the Module**	**No. of Lectures**
1.	Classification of Data	Classification of data, graphic and diagrammatic representation of data, measures of central tendency and dispersion i.e. mean, median, mode and variance, measures of skewness and kurtosis.	8
2.	Permutation and Combination, Probability Theory	Permutations and combinations, Sample space and events, Probability of an event, Axioms of probability, Multiplication and addition theorems, Independent events, Conditional probability, Bayes' theorem.	8

3.	Random Variables	Random Variable, Discrete and continuous distributions, Mean and variance of a random variable, Moments Generating Function (MGF)	4
4.	Probability Distributions	Binomial, Poisson and Normal distributions	5
5.	Statistical Techniques-I	Curve fitting, Method of least squares, Fitting of straight lines, Polynomials, Exponential curves, Correlation and regression.	7
6.	Statistical Techniques-II	Tests of significations: Sampling theory (small & large), Null hypothesis, Alternative hypothesis, testing of hypothesis: Chi-square test, t-test, z-test.	10
		Total number of Lectures	**42**

Evaluation Criteria

Components	**Maximum Marks**
T1	20
T2	20
End Semester Examination	35
TA	25 (Quiz, Assignments, PBL etc.)
Total	**100**

Project Based Learning: Each student in a group of 3-4 students will apply the statistical techniques to solve some real life problems.

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. (Text books, Reference Books, Journals, Reports, Websites etc. in the IEEE format)

1.	**Walpole, R.E, Myers, R.H., Myers S.I and Ye. K**., Probability and Statistics for Engineers and Scientists, 8th Ed., Pearson, 2007
2.	**Papoulis, A. & Pillai, S.U.**, Probability, Random Variables and Stochastic Processes, Tata McGraw-Hill, 2002.
3.	**Spiegel, M.R.,** Statistics (Schaum's oulines), McGraw-Hill, 1995
4.	**Veerarajan, T.,** Probability, Statistics and Random Processes, 3rd Ed. Tata McGraw-Hill, 2008.
5.	**Johnson, R.A.**, Miller and Freund's Probability and Statistics for Engineers, 8th Ed., PHI Learning Private limited, 2011
6.	**Palaniammal, S.,** Probability and Random Processes, PHI Learning Private limited, 2012

CO-PO and CO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2
CO1	2	2	1		2		2	2	
CO2	2	2	2				2	1	
CO3	3	3	2			1	2	2	1
CO4	3	3	3			1	2	2	1
Avg	**2.50**	**2.50**	**2.00**		**2.00**	**1.00**	**2.00**	**1.75**	**1.00**

Detailed Syllabus
Lecture-wise Breakup

Subject Code **NBA Code**	**23B61CA212** **CBAC203**	**Semester:** **(specify Odd/Even):** **Even**	**Semester: 3rd Session:** 2024-25 **Month:** June 2024 to Dec 2024
Subject Name	Object Oriented Programming (C++)		
Credits	**3-1-0**	**Contact Hours**	**4**

Faculty (Names)	**Coordinator(s)**	Dr. Pratik Shrivastava
	Teacher(s) (Alphabetically)	Dr. Pratik Shrivastava, Prof. Sandeep K Singh

COURSE OUTCOMES		**COGNITIVE LEVELS**
CO1	Memorize the fundamental concepts in object-oriented programming and appreciate the differences from other programming paradigms.	Remember (level 1)
CO2	Implement the principles of encapsulation and information hiding effectively in classes and objects.	Apply (level 3)
CO3	Utilize inheritance concepts to build class hierarchies to enhance code reusability and flexibility.	Apply (level 3)
CO4	Demonstrate the use of file and exception handling mechanisms to make code robust, reliable and maintainable code.	Apply (level 3)
CO5	Analyze the need for writing the generic code using templates at class and function level.	Analyze (level 4)

Module No.	**Subtitle of the Module**	**Topics in the Module**	**No. of Lectures for the module**
1	Introduction to Object-Oriented Programming (OOP)	Introduction to Object-Oriented Approach: Overview of Object-Oriented Programming, Comparison with Other Paradigms (Functional, Data Decomposition), Benefits of Object-Oriented Programming, Overview of Programming Paradigms: Imperative Programming, Declarative Programming, Functional Programming, Object-Oriented Programming, Evolution and History of Object-Oriented Programming, Key Concepts of Object-Oriented Programming. Objects and Classes: Encapsulation and Information Hiding, Access Specifiers: Public, Private, Protected Benefits of Encapsulation, Inheritance, Understanding	6

		Inheritance Hierarchies, Types of Inheritance: Single, Multiple, Multilevel Implementing Inheritance in C++ Polymorphism. Overview of C Syntax and Structure: Key Differences Between C and C++, New C++ Constructs: cin, cout, new, delete, Operators, Transition from C to C++. Introduction to C++ Syntax and Features: Differences Between Procedural Programming in C and Object-Oriented Programming in C++, Practical Examples and Exercises Demonstrating C++ Enhancements.	
2	Classes and Objects	Introduction to Classes and Objects: Definition and significance of classes and objects in C++, Real-world analogies to understand classes and objects. Encapsulation and Information Hiding: Concept of encapsulation and its benefits, Access specifiers: public, private, and protected, Examples demonstrating the importance of encapsulation. Abstract Data Types: Understanding abstract data types and their role in OOP, Difference between abstract data types and concrete data types. C++ Class Declaration: Syntax and structure of a C++ class, Declaring attributes (data members) and methods (member functions). State, Identity, and Behavior of Objects: Defining the state, identity, and behavior of objects, How these concepts are represented in C++. Constructors and Destructors: Purpose and types of constructors (default, parameterized, copy), Role of destructors in resource management, Syntax and examples of constructors and destructors. Instantiation of Objects: Creating instances of a class (objects), Understanding the lifecycle of an object. Default Parameter Values: Using default parameters in class methods, Benefits and limitations of default parameter values. Object Types and C++ Garbage Collection: Different types of objects (local, global, static, dynamic), Dynamic memory allocation and deallocation using new and delete. Basic concepts of garbage collection in C++. Abstract Classes: Introduction to abstract classes and their purpose, Declaring and using abstract classes in C++ , Understanding pure virtual functions. Practical Examples and Exercises: Implementing and	10

		practicing the above concepts through coding exercises, Building small projects to reinforce understanding of classes and objects.	
3	Inheritance and Polymorphism	Inheritance: Understanding Class Hierarchy, Types of Inheritance: Public, Private, and Protected, Implementing Inheritance in C++, Aggregation vs. Composition in Inheritance, Resolving Ambiguities in Inheritance. Polymorphism: Introduction to Polymorphism, Categorization of Polymorphism Techniques: Compile-Time and Run-Time, Method Polymorphism: Function Overriding, Polymorphism by Parameter: Function Overloading, Operator Overloading for Polymorphism Implementing Parametric Polymorphism. Examples and Applications: Examples illustrating the use of inheritance to create class hierarchies, Practical applications demonstrating various forms of polymorphism in C++.	10
4	Generic Programming	Template function, function name overloading, Overriding inheritance methods, Run time polymorphism, Multiple Inheritance. Introduction to STL	10
5	Files and Exception Handling	Streams and files, Namespaces, Exception handling,	6
		Total number of Lectures	42

Evaluation Criteria	
Components	**Maximum Marks**
T1	20
T2	20
End Semester Examination	35
TA	25 (Attendance (5), Assignment (5)/Tutorial (5)/Quiz (10))
Total	**100**

Project based learning: Create a C++ application either individually or in groups of maximum 4 students each, to illustrate the concepts covered in class.

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. (Text books, Reference Books, Journals, Reports, Websites etc. in the IEEE format)

	Text Books
1	"Object-Oriented Programming in C++" by Robert Lafore
2	"Programming: Principles and Practice Using C++" by Bjarne Stroustrup
3	"Object-Oriented Programming with C++" by E. Balagurusamy
4	"Problem Solving with C++" by Walter Savitch
	Reference Books
1	"C++ Primer" by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo

2	"Modern C++ Design: Generic Programming and Design Patterns Applied" by Andrei Alexandrescu
3	"Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

CO-PO and CO-PSO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2
CO1	2	2	-	-	-	-	-	2	-
	Understanding object-oriented programming principles requires applying basic knowledge of mathematics, science, and computing fundamentals.	Identifying and understanding problems in object-oriented programming involves analyzing them using basic knowledge and principles.						Proficiency in object-oriented programming contributes to the development of software solutions in AI, ML, Cybersecurity, and Data analytics, aligning with this PSO.	
CO2	-	-	2	1.0	-	-	-	2	-
			Implementing classes and objects is essential for designing and developing computer applications that meet specified needs.	While implementing classes and objects, modern IT tools including database management and networking may be used, but the focus is more on the application design aspect.				Proficiency in implementing classes and objects contributes to the development of software solutions in AI, ML, Cybersecurity, and Data analytics, aligning with this PSO.	
CO3	2	-	2	-	-	-	-	2	-
	Applying inheritance and polymorphism requires a solid understanding of basic object-oriented principles.		Inheritance and polymorphism are essential for designing and developing applications that meet specified needs.					Proficiency in applying inheritance and polymorphism contributes to the development of software solutions in AI, ML, Cybersecurity, and Data analytics,	

								aligning with this PSO.	
CO4	-	-	-	-	1.0	-	1.0	-	0.5
					Effective communication may involve documenting file handling and exception handling processes, but it's not the primary focus of this CO.		Learning file handling and exception handling contributes to lifelong learning and adaptation to technological changes.		While file handling and exception handling are important skills, they are not directly related to full stack web and mobile application development, making the mapping less relevant.
CO5	-	2	-	-	-	1.5	-	2	-
		Analyzing and implementing generic functions and multiple inheritance require problem analysis skills.				While understanding generic functions and multiple inheritance can contribute to project management and teamwork, the direct application is not as significant as other POs.		Proficiency in utilizing generic functions and multiple inheritance contributes to the development of software solutions in AI, ML, Cybersecurity, and Data analytics, aligning with this PSO.	
Average	**2.0**	**2.0**	**2.0**	**1.0**	**1.0**	**1.5**	**1.0**	**2.0**	**0.5**

Detailed Syllabus

Lecture-wise Breakup

Subject Code **NBA Code**	23B61CS217 CBAC204	**Semester: Odd** **(specify Odd/Even)**	**Semester 3rd Session** **Month from July-Dec 2024**
Subject Name	**Digital Logic**		
Credits	3	**Contact Hours**	3-0-0

Faculty (Names)	**Coordinator(s)**	Ms. Anuja Shukla (JIIT 62)
	Teacher(s) (Alphabetically)	**JIIT 62:** Ms. Anuja Shukla, Mr. Rajeev Mishra

COURSE OUTCOMES		**COGNITIVE LEVELS**
CO1	Defining the basics Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, Excess-3.	Remember (Level 1)
CO2	Illustrate the various logic gates and Boolean algebra suite.	Understand (Level 2)
CO3	Identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, Two-Level Circuits.	Understand (Level 2)
CO4	Choose Adders, Subtractors, Multiplexers, Decoders, Encoders, Parity: Binary operations, data manipulation problems.	Apply (Level 3)
CO5	Explain various Flip-Flops, Clocking, Shift Registers, Counters: Sequential circuits, data storage, counting mechanisms.	Analyze (Level 4)
CO6	Determine Counters, Memory, CPU Design, Verilog: Counting, data storage, processing, logic designs and evaluate it.	Evaluate (Level 5)

Module No.	**Subtitle of the Module**	**Topics in the module**	**No. of Lectures for the module**
1.	Number System	Introduction to various number systems: Binary, Octal, Decimal, Hexadecimal, Conversion between different number systems, Binary Coded Decimal (BCD) and its applications, Gray codes and Excess-3 codes	4
2.	Boolean Algebra and Logic Gates	Basic concepts of Boolean algebra, Laws and identities of Boolean algebra, Logic gates: NOT, AND, OR, NAND, NOR, XOR, and XNOR, Universal gates: NAND and NOR, Implementation of logic gates using diodes and transistors	4

3.	Simplification of Boolean Functions	Karnaugh maps for simplification of Boolean expressions, De Morgan's Theorem and its applications, Use of Boolean algebra for logic simplification, Two-level implementations of logic circuits	8
4.	Combinational Logic with MSI and LSI	Adders and subtractors: Half adder, Full adder, half subtractor, Full subtractor, Multiplexers and demultiplexers, Decoders and encoders, Parity generation and checking	09

5.	Sequential Logic	Flip-flops: SR, D, JK, T flip-flops, Clocking in flip-flops: Clocked flip-flops, Master-slave flip-flops, Shift registers: Serial-in-serial-out, Serial-in-parallel-out, Parallel-in-serial-out, Parallel-in-parallel-out, Counters: Ripple counters, Synchronous counters, Modulo counters	06
6.	Registers and Counters	Types of counters: Ring counters, Twisted ring counters, Memory devices: RAM, ROM, PAL, PLA	05
7	Processor Logic Design	Introduction to CPU design, Datapath and ALU design, System design using state machines, Introduction to Verilog for logic design and simulation	06
Total number of Lectures			**42**

Evaluation Criteria

Components	**Maximum Marks**
T1	20
T2	20
End Semester Examination	35
TA	25 (Attendance = (10), Assignments/Mini-Project= (15))
Total	**100**

Project Based Learning: Each student in a group of 2-4 will choose some real-world problems such as This project-based learning experience focuses on immersing students in the fundamental concepts of digital logic systems. Through hands-on projects, students will deepen their understanding of digital circuits, logical operations, and system design principles. real-world problems cover a range of cognitive levels, from basic recall of digital logic concepts to the creation of complex systems that integrate digital logic with other technologies. By addressing these problems, students can enhance their understanding of digital logic systems while developing practical skills applicable to various domains.

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. (Text books, Reference Books, Journals, Reports, Websites etc. in the IEEE format)

1	Author(s): M. Morris Mano, Michael D. Ciletti ,Title: Digital Design ,Edition: 6th Edition ,Publisher: Pearson,Year of Publication: 2020
2	Author(s): Thomas L. Floyd, David M. Buchla ,Title: Digital Fundamentals ,Edition: 12th Edition ,Publisher: Pearson ,Year of Publication: 2019
3	Author(s): Charles H. Roth, Jr., Larry L. Kinney ,Title: Fundamentals of Logic Design Edition: 7th Edition ,Publisher: Cengage Learning,Year of Publication: 2014

CO-PO-PSO Mapping:

COs (NBA Code)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2
C217.1	3	2	3	2	3		3	3	2
C217.2	3	3	3	2	3			3	3
C217.3	2	2	2	2	2			2	2
C217.4				2					
C217.5	3	3	2	2				3	3
C217.6	2	2	2	2	3		1	2	2
NBA Code: C217	2.0	2.0	2.0	2.0	2.0		1.0	2.0	1.0

Justification:

COs	PO1	PO2	PO3	PO4	PO5	PO 6	PO7	PSO1	PSO2
C217.1	3	2	3	2	3		3	2	2
	Apply basic knowledge of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess-3 to provide solutions for complex computer applications using mathematics, science, and computing fundamentals.	Identify, formulate, research literature, and analyze problems in applied computer science using the basics of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess-3, achieving a rating level 2 proficiency.	Design computer applications that meet societal, health, safety, legal, and cultural needs, considering ethics, environment, and sustainability, using the basics of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess-3, achieving a rating level 2 proficiency.	Create, select, and apply appropriate techniques, resources, and modern IT tools, including database management, networking, AI, and ML, using the basics of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess-3, achieving a rating level 2 proficiency.	Communicate effectively through reports, design documentation , and presentations using the basics of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess-3, achieving a rating level 2 proficiency.		Recognize the need for and engage in independent, life-long learning about technological changes in computer applications using the basics of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess-3, achieving a rating level 2 proficiency.	Develop proficiency in software development methodologies and tools to design, implement, and test solutions across AI, ML, Cybersecurity, and Data Analytics using the basics of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess-3, achieving a rating level 2 proficiency.	Develop a versatile skill set with soft skills and programming proficiency in full stack web and mobile application development using the basics of Binary, Octal, Decimal, Hexadecimal, Conversions, BCD, Applications, Gray, and Excess- 3, achieving a rating level 2 proficiency.
C217.2	3	3	3	2	3			2	1
	Apply the basic knowledge of mathematics, science, and computing fundamentals to provide solutions for complex computer applications by illustrating various logic gates and Boolean algebra, achieving a	Identify, formulate, research literature, and analyze problems in applied computer science by illustrating various logic gates and Boolean algebra, achieving a rating level 2 proficiency..	Design computer applications meeting societal, health, safety, legal, and cultural needs with ethical, environmental, and sustainable considerations, integrating various logic gates and Boolean algebra, achieving a	Create, select, and apply modern IT tools, including database management, networking, AI, and ML, with an understanding of their limitations, utilizing knowledge of various logic gates and Boolean	Communicate effective reports, design documentation , and make effective presentations utilizing knowledge of various logic gates and Boolean algebra, achieving a rating level 2 proficiency.			Develop proficiency in software development methodologies and tools for designing, implementing, and testing solutions across AI, ML, Cybersecurity, and Data Analytics, incorporating knowledge of various logic gates and	Develop a versatile skill set encompassing soft skills and programming proficiency in full stack web and mobile application development through illustrating various logic gates and Boolean algebra, achieving a rating level 2 proficiency.

	rating level 2 proficiency.		rating level 2 proficiency.	algebra, achieving a rating level 2 proficiency.				Boolean algebra, achieving a rating level 2 proficiency.	
C217.3	**2**	**2**	2	2	2			**2**	**2**
	Apply the knowledge of mathematics, science, and computing fundamentals to provide solutions for complex computer applications through the identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, and Two-Level Circuits, achieving a rating level 2 proficiency.	Identify, formulate, research literature, and analyze problems in applied computer science through the identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, and Two-Level Circuits, achieving a rating level 2 proficiency.	Design computer applications meeting societal, health, safety, legal, and cultural needs with ethical, environmental, and sustainable considerations, informed by the identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, and Two-Level Circuits, achieving a rating level 2 proficiency.	Create, select, and apply modern IT tools including database management, networking, AI & ML, understanding their limitations, informed by the identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, and Two-Level Circuits, achieving a rating level 2 proficiency.	Communicate effective reports, design documentation, and make effective presentations informed by the identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, and Two-Level Circuits, achieving a rating level 2 proficiency.			Develop proficiency in software development methodologies and tools for designing, implementing, and testing solutions across major core areas of AI, ML, Cybersecurity, and Data Analytics, informed by the identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, and Two-Level Circuits, achieving a rating level 2 proficiency.	Develop a versatile skill set incorporating soft skills and programming proficiency in full stack web and mobile application development through identification and description of various components, architectures of Karnaugh Maps, DE Morgan's Theorem, Boolean Algebra, and Two-Level Circuits, achieving a rating level 2 proficiency.
C217.4				2					
				Create, select, and apply modern IT tools including database management, networking, AI & ML, understanding their limitations, for binary operations and data manipulation problems involving Adders, Subtractors, Multiplexers, Decoders, Encoders, and Parity, achieving a rating level 2 proficiency.					
C217.5	**3**	3	2	2				**1**	**1**
	Apply the knowledge of mathematics, science, and computing fundamentals to provide	Identify, formulate, research literature, and analyze problems in applied	Design computer applications meeting societal, health, safety, legal, and	Apply modern IT tools including database management, networking,				Develop proficiency in software development methodologies and tools for designing,	Develop a versatile skill set incorporating soft skills and programming proficiency in full stack web and

	solutions for complex computer applications through explaining various Flip-Flops, Clocking, Shift Registers, and Counters, which are essential for sequential circuits, data storage, and counting mechanisms, achieving a rating level 2 proficiency.	computer science by explaining various Flip-Flops, Clocking, Shift Registers, and Counters, essential for sequential circuits, data storage, and counting mechanisms, achieving a rating level 2 proficiency.	cultural needs with ethical, environmental, and sustainable considerations, integrating explanations of various Flip-Flops, Clocking, Shift Registers, and Counters crucial for sequential circuits, data storage, and counting mechanisms, achieving a rating level 2 proficiency.	AI & ML, understanding their limitations, for designing sequential circuits, data storage, and counting mechanisms involving Flip-Flops, Clocking, Shift Registers, and Counters, achieving a rating level 2 proficiency.				implementing, and testing solutions across major core areas of AI, ML, Cybersecurity, and Data Analytics, incorporating explanations of various Flip-Flops, Clocking, Shift Registers, and Counters essential for sequential circuits, data storage, and counting mechanisms, achieving a rating level 2 proficiency.	mobile application development, enriched by explanations of various Flip-Flops, Clocking, Shift Registers, and Counters essential for sequential circuits, data storage, and counting mechanisms, achieving a rating level 2 proficiency.
C217.6	**2**	**2**	**2**	**2**	**3**		**1**	**2**	**2**
	Apply the knowledge of mathematics, science, and computing fundamentals to provide solutions for complex computer applications through the determination, evaluation, and application of Counters, Memory, CPU Design, and Verilog for counting, data storage, processing, and logic designs, achieving a rating level 2 proficiency..	Identify, formulate, research literature, and analyze problems in applied computer science through the determination, evaluation, and analysis of Counters, Memory, CPU Design, and Verilog for counting, data storage, processing, and logic designs, achieving a rating level 2 proficiency.	Design computer applications meeting societal, health, safety, legal, and cultural needs with ethical, environmental, and sustainable considerations, leveraging the determination, evaluation, and refinement of Counters, Memory, CPU Design, and Verilog for counting, data storage, processing, and logic designs, achieving a rating level 2 proficiency.	Apply appropriate techniques, resources, and modern IT tools including database management, networking, AI & ML, understanding their limitations, for the determination, evaluation, and refinement of Counters, Memory, CPU Design, and Verilog in counting, data storage, processing, and logic designs.	Communicate effective reports, design documentation, and make effective presentations showcasing the determination, evaluation, and refinement of Counters, Memory, CPU Design, and Verilog for counting, data storage, processing, and logic designs.		Recognize the need for and engage in independent, life-long learning about technological changes in computer applications, demonstrated through the determination, evaluation, and continuous refinement of Counters, Memory, CPU Design, and Verilog for counting, data storage, processing, and logic designs, achieving a rating level 2 proficiency.	Develop proficiency in software development methodologies and tools for designing, implementing, and testing solutions across major core areas of AI, ML, Cybersecurity, and Data Analytics, utilizing skills in determining Counters, Memory, CPU Design, and Verilog for counting, data storage, processing, and logic designs, achieving a rating level 2 proficiency.	Develop a versatile skill set encompassing soft skills and programming proficiency in full stack web and mobile application development, enriched by determining Counters, Memory, CPU Design, and Verilog for counting, data storage, processing, and logic designs, achieving a rating level 2 proficiency.
Avg	**2.0**	**2.0**	**2.0**	**2.0**	**2.0**		**1.0**	**2.0**	**1.0**

Course Description

Course Code	**23B31HS211**	**Semester Odd 2024**	**Semester: III Session: 2024-25 Month from Jul to Dec 2024**
Course Name	**Soft Skills II**		
Credits	**2**	**Contact Hours**	**2-0-0**
Faculty (Names)	**Coordinator(s)**	Dr Paridhi Chaudhary	
	Teacher(s) (Alphabetically)		
COURSE OUTCOMES: After the successful completion of this course, the student will be able to			**COGNITIVE LEVELS**
CO1	Understand the basic concepts of Language and Literature and relate their concepts with each other.		Understanding (C1)
CO2	Apply various employment related communication skills to enhance their employability.		Applying (C2)
CO3	Apply various communication skills through verbal and non-verbal communication.		Applying (C3)
CO4	Discover the familiarity with diverse forms of literary skills to be able to analyse the various facets of life.		Analysing (C4)
Module No.	**Title of the Module**	**Topics in the Module**	**No. of Lectures**
1.	Introduction to Language and Literature	Definition of Language, its Characteristics and function. What is Literary Language? And its use in Literature. Its Uses and Rationale in Soft Skills; Characteristics of Literature, Recent Perspectives; Rhetorical Devices.	6
2.	Academic Skills	**Employment Communication:** Introduction, Resume, Curriculum Vitae, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter, Professional Presentation: Nature of Oral Presentation, planning a Presentation, Preparing the Presentation, Delivering the Presentation. **Job Interviews:** Introduction, Importance of Resume, Definition of Interview, Background	6

		Information, Types of Interviews, Preparatory steps for Job Interviews, Interview Skills, Changes in the Interview Process, FAQ during Interviews. **Group Discussion:** Introduction, Ambience/ Seating Arrangement for Group Discussion, Importance of Group Discussions, Difference between Group Discussion.	
3.	Reading and Listening Skills	Types of reading– pleasure/ info/knowledge Reading Strategies- predicting, skimming, scanning, reading intensively Active Reading – understanding, highlighting, making notes. Listening effectively and identifying barriers Listening for specific and general information	8
4.	Writing Skills	Purpose of writing, Clarity in Writing, Principle of Effective writing, Writing Techniques. Business Letters and Reports: Need and functions of Business Letters: Planning & Layout of Business Letters- Kinds of Business Letters- essentials of effective correspondence, purpose, kind and objective of reports, Writing Reports	6
5	Assertive Skills	*If* by Rudyard Kipling *The Grasshoper and the Ant* – Jean De La Fontaine	2
		Total number of Lectures	**28**

Evaluation Criteria **Components**	**Maximum Marks**
Mid Term	30
End Semester Examination	40
TA	30 (Quiz, Assignments, Project, Class Attendance &

Participation) **Total** **100**	
PBL Component: The project is to be done in a group of 3-4 students. Students will be asked to write a proposal with a well-researched technical report on any social issue.	
Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. (Text books, Reference Books, Journals, Reports, Websites etc)	
1.	C.L. Bovee, J.V.Thill, and M.Chaturvedi, *Business Communication Today*,9th Ed, Pearson Education, copyright@ Dorling Kinderslay (India) Pvt Ltd,2009
2.	K. M. Quintanilla and S.T.Wahl, *Business and Professional Communication*, Sage Publications Pvt India Ltd,2011
3.	S. Kumar, P. Lata, *Communication Skills*, Oxford University Press,1st, Ed. 2011
4.	R.K Bansal, J.B Harrison, *Spoken English for India*, Orient Longman, 2018
5.	M A Yadugiri, The Pronunciation of English: Principles and Practice, Viva Books Pvt. Ltd, India, 2015
6.	A. R. Rizvi, Effective Technical Communication, 2nd edition, McGraw Hill Education Private Limited, Chennai, 2018.
7.	R. Murphy, English Grammar in Use, 4th edition, Cambridge University Press, 2012.
8.	M. Hewings, *English Pronunciation in Use*. Advanced. Cambridge: CUP, 2009
9.	K. Mohan, N. P. Singh, *Speaking English Effectively* 2nd Edition. Macmillan Publishers India Ltd. Delhi. 2011
10.	E. S. Kumar, P. Sreehari, *A Handbook for English Language Laboratories*. New Delhi: Foundation, 2009.

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PSO1**	**PSO2**
CO1					1	1					
CO2					1	1	1				1
CO3											1
CO4					1		1				1
Avg					1	1	1				1

Detailed Syllabus
Lecture-wise Breakup

Course Code **NBA Code**	23B61CA213 CBAC206	**Semester:** ODD	**Semester:** III **Session:** 2024-25 **Month from** July'24 **to** Dec'24
Course Name	Multimedia Technology-II		
Credits	2-0-0	**Contact Hours**	2

Faculty (Names)	**Coordinator(s)**	Mr. Noor Mohammad
	Teacher(s) (Alphabetically)	Dr. Diksha Chawla, Mr. Noor Mohammad

COURSE OUTCOMES		**COGNITIVE LEVELS**
CO1	Describe the necessity of data compression in multimedia.	Understand (Level 2)
CO2	Interpret widely used multimedia compression standards and their real-world applications.	Apply (Level 3)
CO3	Illustrate audio-video processing and transmission.	Apply (Level 3)
CO4	Compare the technical aspects of various wired and wireless networking protocols in multimedia.	Analyze (Level 4)
CO5	Summarize the effectiveness of multimedia data compression algorithms for a given real-world problem statement.	Evaluate (Level 5)

Module No.	**Title of the Module**	**Topics in the Module**	**No. of Lectures for the module**
1	Multimedia Data Compression	Data Compression, Need of Data Compression in Multimedia, Compression Techniques: Lossless and Lossy Compression, Information Theory, Compression Ratio.	2
2	Data Compression Techniques- I	Compression: Lossy and lossless compression algorithms. Entropy Encoding: Run Length Encoding, Huffman Encoding, Arithmetic Encoding and source encoding.	5
3	Data Compression Techniques- II	DCT, DFT, Text Compression: Static Huffman Encoding. Image Compression: JPEG, LZW.	6
4	Digital Audio	Digitization, Nyquist Theorem, Signal-to-Noise Ratio (SNR), quantization and transmission of audio, basic audio compression methods.	5
5	Fundamentals of Video	Analog video: NTSC, PAL and SECAM video. Digital video: Chroma Subsampling, HDTV, UHDTV, 3D video. Video Color Models. Video compression based on motion compensation: MPEG-1, MPEG-2, MPEG-4, MPEG-7 and MPEG-21.	7
6	Multimedia Communications	Internet Technologies and Protocols: IP, TCP, UDP, Firewall, QoS, HTTP, Real-Time Transport Protocol, CDN, Broadcast/Multicast. Wireless Network Technologies: 1G, 2G, 3G, 4G.	5
		Total number of Lectures	**30**

Evaluation Criteria

Components	**Maximum Marks**
Mid Term Examination	30
End Term Examination	40
TA	
1. Attendance	05
2. Assignments (15), PBL (10)	25
Total	**100**

Project based learning: Report the effectiveness, efficiency and proficiency of multimedia data compression standards on a real-world problem in groups of maximum 3 students each, to illustrate the concepts covered in class.

Recommended Textbooks: Author(s), Title, Edition, Publisher, Year of Publication etc.

1.	Z.-N. Li, M. S. Drew, and J. Liu, *Fundamentals of multimedia*, 3rd ed. Cham, Switzerland: Springer Nature, 2022.
2.	R. Steinmetz and K. Nahrstedt, *Multimedia: Computing, Communications and Applications*. Pearson, 2012.

Recommended Reference Books: Author(s), Title, Edition, Publisher, Year of Publication etc.

1.	Khalid Sayood, *Introduction to data compression*. Cambridge, Ma: Mogan Kaufmann, 2018.
2.	R. Steinmetz and K. Nahrstedt, *Multimedia Fundamentals, Volume 1: Media Coding and Content Processing*. Pearson Education, 2002.
3.	O. Kalipsiz, 'Multimedia databases', in 2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics, London, UK, 2002.
4.	K. R. Rao and J. J. Hwang, *Techniques & Standards for Image, Video & Audio Coding*. New Jersey: Prentice Hall, 1996.

CO-PO-PSO Mapping									
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	2	2					1		
	Moderately able to analyze the mathematics and computing behind data compression.	Moderately able to analyze the problems based on data compression.					Able to develop a mindset towards continuous learning and adaptation.		
CO2	3	2	1				2		2
	Able to apply appropriate compression standards in real-world scenarios.	Moderately able to choose appropriate compression algorithms on given problems.	Slightly able to design data compression in real-world applications.				Able to stay updated with the latest advancements in compression standards.		Moderately able to use compression standards in web/app development.
CO3	3	2	1				2		2
	Able to effectively explain the mechanisms behind audio-video processing and transmission.	Moderately able to identify and analyze problems of audio-video processing.	Slightly able to design audio-video processing in real-world applications.				Audio-video technologies evolve rapidly, requiring continuous learning to stay updated.		Moderately inculcate audio-video processing in web/app development.
CO4	3	2	1				2		2
	Able to analyze the strengths and weaknesses of wired versus wireless protocols.	Moderately able to analyze the technical aspects of networking protocols.	Slightly able to design networking protocols in real-world applications				Moderately able to stay updated with recent networking protocols.		Moderately able to utilize networking protocols in web/app development.
CO5	2	2			2	2	2		
	Moderatelyable to perform statistical analysis to compare algorithms.	Moderately able to report the effectiveness of compression algorithms in a given problem..			Moderately report the effectiveness, efficiency and proficiency of data compression standards.	Moderately able to handle a team.	Reporting requires continuous learning and assessment.		
Avg.	**2.6**	**2**	**1**		**2**	**2**	**1.8**		**2**

Detailed Syllabus
Lab Session-wise Breakup

Subject Code **NBA Code**	23B65CA214 CBAC251	**Semester: Odd** **(specify Odd/Even)**	**Semester Session 2024-25** Month from: July to December 2024
Subject Name	Algorithm and Problem Solving Lab		
Credits	**0-0-1**	**Contact Hours**	**2**

Faculty (Names)	**Coordinator(s)**	Dr. Meenal Jain
	Teacher(s) (Alphabetically)	Dr. Meenal, Dr. Shelendra Pal, Dr. Shobhit Tyagi

COURSE OUTCOMES		**COGNITIVE LEVELS**
CO1	Understand complexity using asymptotic and experimental analysis for various algorithmic design techniques.	Understand Level (Level 2)
CO2	Practice standard algorithms for searching and sorting	Apply Level (Level 3)
CO3	Apply and build various algorithms and design techniques to solve the given problem.	Apply Level (Level 3)
CO4	Formulate, elaborate and design an efficient solution to a given problem using appropriate data structure and algorithm design technique	Evaluate Level (Level 5)

Module No.	Title of the Module	List of Experiments	No. of Labs for the module
1.	Analysis of algorithms, Searching and sorting based problems	Introduction to problem solving approach; Asymptotic Analysis; Solving Recurrences; Empirical analysis of sorting and searching algorithms – Merge sort, Quick sort, Heap sort, Radix sort, Count sort, Binary search, and Median search	4
2.	Design Technique: Divide and Conquer	Problems based on Divide and Conquer (D&C) approach such as Binary search, Quick sort, and Merge sort; and Closest pair, etc.	2
3.	Design Technique: Backtracking Algorithms	Review of backtracking based solution approach using N queen, and Rat in a maze; Travelling salesman problem	2
4.	Design Technique: Greedy Algorithms	Introduction to greedy based solution approach; Minimum Spanning Trees (Prim's and Kruskal algorithms); Shortest Path using Dijkstra's algorithm; Fractional and 0/1 Knapsack; Coinage problem;	2
5.	Dynamic Programming	Fundamentals of Dynamic programming based solution approach; 0/1 Knapsack; Shortest path using Floyd Warshall; Coinage problem; Longest common subsequence; Longest increasing sequence,	2
6.	Problem solving by search	Uninformed search (BFS, DFS)	2
		Total number of Lab	14

Evaluation Criteria

Components	**Maximum Marks**
Eval 1	15
Eval 2	15
Lab Test 1	20
Lab Test 2	20
PBL	20 (Students will submit the mini project in a group of 3-4 members)
Attendance	10
Total	**100**

Project based learning: Students in a group of 3-4 will be designing an efficient solution to a given problem / case-studies using appropriate data structure and algorithm design technique studies in the course. The students have to implement the mini project using C/C++/Java language. Project development and its presentation will enhance coding skills, knowledge and employability of the students in IT sector.

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. **(Reference Books,** Journals, Reports, Websites etc. in the IEEE format)

1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, MIT Press, 3rd Edition, 2009
2.	Steven Skiena ,The Algorithm Design Manual, Springer; 2nd edition , 2008
3.	Knuth, The art of Computer Programming Volume 1, Fundamental Algorithms, Addison-Wesley Professional; 3 edition,1997
4.	Horowitz and Sahni, Fundamentals of Computer Algorithms, Computer Science Press, 2008

5.	Sedgewick, Algorithms in C, 3rd edition. Addison Wesley, 2002
6.	Alfred V. Aho, J.E. Hopcroft, Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley Series in Computer Science and Information Processing, 1983
7.	ACM Transactions on Algorithms (TALG)
8.	Algorithmica Journal, Springer
9.	Graphs and Combinatorics, Journal, Springer
10.	The ACM Journal of Experimental Algorithmics

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. **(Text books)**	
1.	T. Roughgarden, *Algorithms Illuminated: Part 1: The Basics*, 1st ed., Soundlikeyourself Publishing, Sept. 2017.
2.	.T. Roughgarden, *Algorithms Illuminated: Part 2: Graph Algorithms and Data Structures*, 1st ed., Soundlikeyourself Publishing, 2018.
3.	T. Roughgarden, *Algorithms Illuminated: Part 3: Greedy Algorithms and Dynamic Programming*, 1st ed., Soundlikeyourself Publishing, 2019.
4.	M. A. Weiss, *Data Structures and Algorithm Analysis in C++*, 4th ed., Pearson, 2014.

CO Code	PO1	PO2	PO3	PO4	PO 5	PO6	PO7	PSO1	PSO2
CO1	1	3	2			2	1	1	2
	Slightly related to understanding the logic behind algorithm design	Strongly mapped to the analysis of algorithms.	Moderately mapped to understanding designing of algorithm.			Moderately mapped as to the analysis of algorithms.	Slightly mapped as analysis of algorithm is used throughout their life as software developers.	slightly Mapped as understanding of fundamentals of algorithm in real world applications	Moderately mapped to algorithm design techniques as it is going to be used everywhere
CO2		1	3			2	1	1	1
		Slightly related to understanding the concepts of searching and sorting.	Strongly Mapped as understanding of fundamentals of searching and sorting for given problem.			Moderately related to understanding the concepts of searching and sorting.	Slightly mapped as searching and sorting concepts learned are always used.	Slightly Mapped as understanding of fundamentals of searching and sorting in real world applications	Slightly Mapped as understanding of basics of searching and sortings help to develop real world projects and solutions
CO3		1	3			2	1	1	1
		Slightly mapped to analysis of designed algorithm	Strongly mapped to development of solution.			Moderately mapped to usage of design techniques in individually and as a team.	Slightly mapped to use of design techniques which will help them life- long	Slightly Mapped as design of algorithm for real world applications	Slightly mapped to use of algorithm design techniques in different competitions
CO4		2	2			3	2	1	2

		Moderately mapped to evaluation of problem	Moderately mapped to designing and analysis of efficient solution.			Strongly mapped to development of solution as a team	Moderately mapped to lifelong learning	Slightly Mapped as design and analysis of algorithm is used to for real-life applications.	Moderately mapped to use project development experience.
Average	**1.00**	**1.75**	**2.50**			**2.25**	**1.25**	**1.00**	**1.50**

Detailed Syllabus
Lab-wise Breakup

Course Code **NBA Code**	23B65CA215 CBAC252	**Semester Odd** **(specify Odd/Even)**	**Semester III Session** 2024-25 **Month from** July. to Dec. 2024
Course Name	**Object Oriented Programming (C++) Lab**		
Credits	0-0-1	**Contact Hours**	2hrs

Faculty (Names)	**Coordinator(s)**	Mr. Mohit Singh
	Teacher(s) (Alphabetically)	Mr. Mohit Singh, Dr. Pratik Shrivastava, Ms. Ritika, Prof. Sandeep K Singh

COURSE OUTCOMES		**COGNITIVE LEVELS**
CO1	Use Basic OOP Concepts and C++ Syntax	Applying Level(C3)
CO2	Implement Data Abstraction, Inheritance, and Polymorphism	Applying Level(C3)
CO3	Utilize Virtual Functions to Achieve Polymorphism	Applying Level (C3)
CO4	Apply Generic Programming and Utilize the Standard Template Library (STL)	Applying Level (C3)
CO5	Implement Robust Exception Handling Mechanisms	Applying Level(C3)

Module No.	**Title of the Module**	**Topics in the Module**	**No. of Labs for the module**
1.	**Introduction to Object-Oriented Programming (OOP) and C++ Basics**	Overview of OOP Concepts and Importance in Software Development, Basic Syntax and Structure of C++ Programming Language, Understanding Data Types, Variables, and Control Flow Statements	3
2.	**Principles of Data Abstraction, Inheritance, and Polymorphism**	Exploring Data Abstraction and its Benefits, Implementing Inheritance in C++, Application of Polymorphism in Object-Oriented Programming	3
3.	**Virtual Functions and Polymorphism**	Introduction to Virtual Functions and their Concept, Achieving Polymorphism through Function Overriding, Utilizing Polymorphism for Flexible Software Design	2

4.	**Generic Programming and STL**	Usage of class templates and function templates, Introduction to STL- containers, iterators, and functions	3
5.	**Input/Output Operations in C++**	Managing Formatted I/O using Stream Classes, Handling Unformatted I/O with Stream Manipulators, Practical Examples, and Exercises on Input/Output Operations	1
6.	**Exception Handling in C++**	Introduction to Exceptions and Error Handling, Implementation of Try-Catch Blocks for Error Management, Ensuring Robustness through Effective Exception-Handling Techniques	2
		Total number of Labs	**14**

Evaluation Criteria

Components	**Maximum Marks**
Evaluation 1	**15**
Lab Test1	**20**
Evaluation 2	**15**
Lab Test 2	**20**
Mini Project (PBL)	**10**
Attendance	**10**
TA	**10**
Total	**100**

Project based leaning: Groups of 3-4 students will choose a project topic. They will use the concepts of OOP to execute their project. In a team, they will learn how to apply the concepts for problem solving in a meaningful way.

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. (Text books, Reference Books, Journals, Reports, Websites etc. in the IEEE format)

1	Herbert Schildt, C++: The Complete Reference, McGraw-Hill Osborne Media, 4th Edition, 2017
2	Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition, 2016
3	Stroustrup B., The C++ Programming Language, Addison Wesley, 4th Edition, 2013
4	Avi Silberschatz, Henry F. Korth, and S. Sudarshan, “Database System Concepts”, 6th edition, McGrawHill, 2010.

5	Robert Lafore, Object Oriented Programming in C++, SAMS, 4th Edition, 2002
6	John Hubbard, Schaum's Outline of Programming with C++, McGraw-Hill, 2nd Edition, 2000

CO/PO Mapping:

COs/POs/PSOs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1: Understand and Apply Basic OOP Concepts and C++ Syntax	**3**	**2**	**2**	**3**	**2**	**1**	**2**	**3**	**3**
CO2: Implement Data Abstraction, Inheritance, and Polymorphism	**3**	**3**	**3**	**3**	**2**	**2**	**3**	**3**	**3**
CO3: Utilize Virtual Functions to Achieve Polymorphism	**3**	**3**	**3**	**3**	**2**	**2**	**3**	**3**	**3**
CO4: Apply Generic Programming and Utilize the Standard Template Library (STL)	**3**	**3**	**3**	**3**	**2**	**2**	**3**	**3**	**3**
CO5: Implement Robust Exception Handling Mechanisms	**3**	**3**	**3**	**3**	**2**	**2**	**3**	**3**	**3**

- **CO1: Understand and Apply Basic OOP Concepts and C++ Syntax**
  - **PO1**: Strong correlation (3) as it involves applying fundamental computing concepts.
  - **PO2**: Moderate correlation (2) since it involves basic problem analysis.
  - **PO3**: Moderate correlation (2) for designing simple applications.
  - **PO4**: Strong correlation (3) as it involves using modern programming tools.
  - **PO5**: Moderate correlation (2) for communicating simple designs.
  - **PO6**: Low correlation (1) as it involves individual work.
  - **PO7**: Moderate correlation (2) for continuous learning of programming.
  - **PSO1**: Strong correlation (3) for proficiency in software development.
  - **PSO2**: Strong correlation (3) for foundational skills in development.
- **CO2: Implement Data Abstraction, Inheritance, and Polymorphism**
  - **PO1**: Strong correlation (3) as it involves advanced computing concepts.
  - **PO2**: Strong correlation (3) due to the need for detailed problem analysis.
  - **PO3**: Strong correlation (3) for designing more complex applications.
  - **PO4**: Strong correlation (3) as it involves advanced tool usage.
  - **PO5**: Moderate correlation (2) for explaining complex designs.

- **PO6**: Moderate correlation (2) for teamwork and project management.
- **PO7**: Strong correlation (3) for ongoing learning in advanced concepts.
- **PSO1**: Strong correlation (3) for software development methodologies.
- **PSO2**: Strong correlation (3) for advanced programming proficiency.

- **CO3: Utilize Virtual Functions to Achieve Polymorphism**

  - **PO1**: Strong correlation (3) as it involves advanced computing concepts.
  - **PO2**: Strong correlation (3) due to the need for detailed problem analysis.
  - **PO3**: Strong correlation (3) for designing polymorphic applications.
  - **PO4**: Strong correlation (3) as it involves advanced tool usage.
  - **PO5**: Moderate correlation (2) for explaining polymorphism.
  - **PO6**: Moderate correlation (2) for teamwork and project management.
  - **PO7**: Strong correlation (3) for ongoing learning in advanced concepts.
  - **PSO1**: Strong correlation (3) for software development methodologies.
  - **PSO2**: Strong correlation (3) for advanced programming proficiency.

- **CO4: Apply Generic Programming and Utilize the Standard Template Library (STL)**

  - **PO1**: Strong correlation (3) as it involves advanced computing concepts.
  - **PO2**: Strong correlation (3) for problem analysis using templates.
  - **PO3**: Strong correlation (3) for designing applications with STL.
  - **PO4**: Strong correlation (3) as it involves modern tool usage.
  - **PO5**: Moderate correlation (2) for explaining generic programming.
  - **PO6**: Moderate correlation (2) for teamwork and project management.
  - **PO7**: Strong correlation (3) for ongoing learning in advanced concepts.
  - **PSO1**: Strong correlation (3) for software development methodologies.
  - **PSO2**: Strong correlation (3) for advanced programming proficiency.

- **CO5: Implement Robust Exception Handling Mechanisms**

  - **PO1**: Strong correlation (3) as it involves fundamental computing concepts.
  - **PO2**: Strong correlation (3) for analyzing error conditions.
  - **PO3**: Strong correlation (3) for designing robust applications.
  - **PO4**: Strong correlation (3) as it involves modern error-handling tools.
  - **PO5**: Moderate correlation (2) for communicating error handling.
  - **PO6**: Moderate correlation (2) for teamwork in error handling.
  - **PO7**: Strong correlation (3) for ongoing learning in advanced concepts.
  - **PSO1**: Strong correlation (3) for software development methodologies.
  - **PSO2**: Strong correlation (3) for advanced programming proficiency.

Detailed Syllabus
Lab-wise Breakup

Course Code **NBA Code**	23B65CA216 CBAC253	**Semester:** ODD	**Semester:** III **Session:** 2024-25 **Month:** July'24 **to** December'24
Course Name	Multimedia Technology-II Lab		
Credits	0-0-1	**Contact Hours**	2

Faculty (Names)	**Coordinator(s)**	Dr. Sayani Ghosal
	Teacher(s) (Alphabetically)	Dr. Diksha Chawla, Mr. Noor Mohammad, Ms. Purtee J Kohli, Dr. Sayani Ghosal

COURSE OUTCOMES		**COGNITIVE LEVELS**
CO1	Practice using advanced tools and features to design complex illustrations and publications.	Apply (Level 3)
CO2	Employ image processing libraries in Python for usage in different applications.	Apply (Level 3)
CO3	Design visually appealing infographics for effective communication.	Create (Level 6)
CO4	Develop branding assets that are engaging and impactful.	Create (Level 6)

Module No.	**Title of the Module**	**List of Experiments**	**No. of Lectures**
1	Adobe Illustrator-I	Layers, Group and Ungroup, Modify Shapes, Pathfinder, Rotate and Reflect Tools, Knife, Eraser and Scissor Tool, Envelope Distort.	2
2	Adobe Illustrator - II	Pen tool and Anchor Points, Grid, Guide and Brush Tools, Blend Tool, Creation of 3D objects, Raster to Vector Conversion, Create Graph. Design Infographics.	3
3	Image Processing using Python	Overview of OpenCV, Read and Display Images, Image Formats and Conversions, Color Space Conversion, Histogram Equalization, Geometric Transformations.	4
4	Brand Assets	Logo Design, Typography in Logo Design, Business card Design, Blog Design, Social Media Post Design, Brochure Design.	5
5	Adobe Creative Cloud	Overview of Creative Cloud suite.	1
		Total number of lectures	15

Evaluation Criteria

Components	**Maximum Marks**
Evaluation 1	15
Lab Test-1	20
Evaluation 2	15
Lab Test-2	20
PBL (15), Lab Assignments (5)	20
Attendance	10
Total	**100**

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. (Text books, Reference Books, Journals, Reports, Websites etc. in the IEEE format) **Text Book**	
1	Brian Wood, Adobe Illustrator Classroom in a Book (2023 release): The Official Training Workbook from Adobe, Adobe Press, 1st Edition, 17 February 2023
2	S. Dey, Hands-On Image Processing with Python (2018). Packt Publishing, 1st Edition, 30 November 2018
Reference Material	
1	"Adobe Illustrator Help." [Online]. Available: https://helpx.adobe.com/pdf/illustrator_reference.pdf

CO-PO-PSO Mapping									
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1		2	2	3	1		2		2
		Moderately identify, formulate, and analyze problems in complex illustration.	Moderately able to design illustration.	Apply appropriate tool to design illustration.	Slightly contribute to make effective communication		Moderately able to engage in independent and life-long learning		Moderately inculcate soft skills to design illustration
CO2		2	1	2			1	1	1
		Moderately identify and analyze problems related to image.	Slightly able to contribute in application development.	Moderately able to perform Image Processing			Slightly able to engage in independent and life-long learning	Slightly contribute to AI ML	Slightly develop programming proficiency
CO3		2	2	3	2	1	2		1
		Moderately able to identify and formulate visual elements.	Moderately contribute to design Infographics.	Apply appropriate visual elements to design Infographics.	Moderately contribute to make effective communication	Slightly able to work with team	Moderately able to engage in independent and life-long learning		Slightly contribute Infographics in web development.
CO4			2	3	2	1	2		1
			Moderately contribute to design branding assets.	Apply appropriate tool to design branding assets.	Moderately contribute to make effective communication	Slightly able to work with team	Moderately able to engage in independent and life-long learning		Slightly inculcate soft skills to design branding assets.
Avg.		2	1.75	2.75	1.67	1	1.75	1	1.25