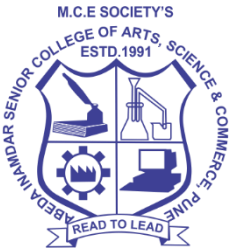

M.C.E. Society's

ABEDA INAMDAR SENIOR COLLEGE OF ARTS, SCIENCE AND COMMERCE (AUTONOMOUS), PUNE

AZAM CAMPUS, CAMP, PUNE – 411001

Syllabus of S.Y. B.S.c (Computer Science)

Applicable for the Autonomous College affiliated to

Savitribai Phule Pune University

B.S.c(Computer Science Honours) Four Year Degree Programme (Choice Based Credit System)

(NEP 2023 Pattern)

With effect from June 2024

SEMESTER III				
Course Type	**Course Name**	**Credits**		
		Theory	**Practical**	**Total**
Major/Core Theory	Data Structures and Algorithms-I	2		
Major/Core Theory	Python Programming	2		
Major/Core Theory	Software Engineering	2		
Major/Core Practical	Lab Course in Python Programming		2	
Minor	Mathematics for Artificial Intelligence- II OR Statistics for Data Science-II	2		
Minor Practical	Mathematics Practical –II OR Statistics Practical –II		2	
GE/OE	Cyber Law-I	2		
Vocational Skill Course	Lab Course in Data Structures and Algorithms-I		2	
AECC	Hindi	2		
Co-Curricular Courses	Basics of Yoga		2	
Field Project	Field Project based on Software Engineering		2	
	Total	12	10	22

SEMESTER IV				
Course Type	**Course Name**	**Credits**		
		Theory	**Practical**	**Total**
Major/Core Theory	Data Structures and Algorithms-II	2		
Major/Core Theory	Object Oriented Programming using C++	2		
Major/Core Theory	Evolution of Network and Data Communication	2		
Major/Core Practical	Lab Course in Object Oriented Programming using C++		2	
Minor	Mathematics for Artificial Intelligence -III OR Statistics for Data Science- III	2		
Minor Practical	Mathematics Practical –III OR Statistics Practical –III		2	
GE/OE	Cyber Law-I I	2		
SEC	Lab Course in Data Structures and Algorithm s-II		2	
AECC	Hindi	2		
Co-Curricular Courses	Yoga & Wellness	2		
CEP	Survey & Interpretation		2	
	Total	14	8	22

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Data Structures and Algorithms – I**
Course Code	**23SBCS31MM**
Semester	**III**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	To learn the systematic way of solving problem.
2.	To understand the different methods of organizing large amount of data.
3.	To efficiently implement the different data structures.
4.	To efficiently implement solutions for specific problems.
5.	To apply linear data structures.

Course Outcome	
1.	To use well-organized data structures in solving various problems.
2.	To differentiate the usage of various structures in problem solution.
3.	Implementing algorithms to solve problems using appropriate data structures.

Syllabus		
Unit I	**Introduction to Data Structures and Algorithm Analysis**	**4 Hours**
	1. Introduction 1.1 Need of Data Structure 1.2 Definitions - Data and information, Data type, 1.3 Data object, ADT, Data Structure 1.4 Types of Data Structures 2. Algorithm analysis 2.1 Space and time complexity 2.2 Best, Worst, Average case analysis, 2.3 Asymptotic notations (Big O, Omega Ω, Theta 2.4 Problems on time complexity calculation.	
Unit II	**Array as a Data Structure**	**10 Hours**
	1. ADT of array, Operations 2. Array applications - Searching 3. Linear search 4. Binary Search 5. Sorting Terminology- Internal, External, Stable, In-place Sorting 6. Sorting Algorithms 7. Bubble Sort, Insertion Sort, Selection Sort 8. Divide and Conquer strategy: Merge Sort, Quick Sort.	
Unit III	**Linked List**	**8 Hours**
	1. List as a Data Structure, differences with array. 2. Dynamic implementation of Linked List, internal and external pointers 3. Types of Linked List – Singly, Doubly, Circular 4. Operations on Linked List - create, traverse, insert, delete, search, sort, reverse, Concatenate, merge, time complexity of operations. 5. Applications of Linked List – polynomial representation, Addition of two polynomials 6. Generalized linked list – concept, representation, multiple-variable polynomial, Representation using generalized list.	
Unit IV	**Stack**	**4 Hours**
	1. Introduction 2. Operations – init(), push(), pop(), isEmpty(), isFull(), peek(). 3. Implementation- Static and Dynamic with comparison 4. Applications of stack 4.1 Function call and recursion, String reversal, palindrome checking 4.2 Expression types - infix, prefix and postfix, expression conversion and evaluation (implementation of infix to postfix, evaluation of postfix)	
Unit V	**Queue**	**4 Hours**
	1. Introduction 2. Operations - init(), enqueue(), dequeue(), isEmpty(), isFull(), peek(),differences with stack. 3. Implementation - Static and Dynamic with comparison 4. Types of Queue - Linear Queue, Circular Queue, Priority Queue, Double Ended Queue(with implementation) 5. Applications – CPU Scheduling in multiprogramming environment,	

Suggested Readings	
1.	Classic Data Structures-D. Samanta, Prentice Hall India Pvt. Ltd.
2.	Fundamentals of Data Structures in C- Ellis Horowitz, SartajSahni,Susan AndersonFreed, 2nd Edition, Universities Press
3.	Data Structures using C and C++-YedidyahLangsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Pearson Education
4.	Data Structures: A Pseudo code approach with C, Richard Gilberg ,Behrouz A. Forouzan, Cengage Learning.
5.	Introduction to Data Structures in C-Ashok Kamthane, Pearson Education
6.	Algorithms and Data Structures, Niklaus Wirth, Pearson Education

Website Reference Links	
1.	https://www.programiz.com/dsa#google_vignette
2.	https://www.geeksforgeeks.org/data-structures/
3.	https://www.codewithharry.com/videos/data-structures-and-algorithms-in-hindi-1/#google_vignette
4.	https://www.tutorialspoint.com/dsa_using_c/index.htm

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1
(Autonomous) Affiliated to Savitribai Phule
Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Python Programming**
Course Code	**23SBCS32MM**
Semester	**III**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	To introduce programming concepts using python.
2.	To develop Programming logic using python.
3.	To develop basic concepts and terminology of python programming
4.	To test and execute python programs

Course Outcome	
1.	Students should be able to develop logic for problem solving
2.	Students should be able to determine the methods to create and develop Python programs by utilizing the data
3.	Student should understand the structures like lists, dictionaries, tuples and sets.
4.	Students should be able to write python programs and develop a small application project

Syllabus		
Unit I	**An Introduction to Python**	**6 Hours**
	1. Introduction to Python -History, features, Applications, Installing Python, Running Simple Python program 2. Basics of Python- Standard data types - basic, none, Boolean (true & False), numbers, Variables, Constants, Python identifiers and reserved words, Lines and indentation, multi-line statements and Comments 3. Input/output with print and input functions Operators- assignment, arithmetic, relational, logical and bitwise operations.	
Unit II	**Control Statement**	**8 Hours**
	1. Sequence Control – Precedence of operators, Type conversion 2. Conditional Statements: if, if-else, nested if-else 3. Looping- for, while, nested loops, loop control statements(break, continue, pass) Strings: declaration, manipulation, special operations, escape character, string formatting operator, Raw String,Unicode strings, Built-in String methods.	
Unit III	**Lists, functions, tuples and dictionaries, Sets**	**10 Hours**
	1. Python Lists: Concept, creating and accessing elements, updating & deleting lists, traversing a List, reverse Built-in List Operators, Concatenation, Repetition, In Operator, Built-in List functions and methods. 2. Functions: Definitions and Uses, Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Flow of Execution, Parameters and Arguments, Variables and Parameters, Stack Diagrams, Void Functions, Anonymous functions Importing with from, Return Values, Boolean Functions, More Recursion 3. Functional programming tools - filter (), map (), and reduce (), recursion, lambda forms. 4. Tuples and Dictionaries: Tuples, Accessing values in Tuples, Tuple Assignment, and Tuples as return values, Variable-length argument tuples, and Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built-in tuple functions, indexing, slicing and matrices. Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, and Built-in Dictionary Methods. 5.Sets- Definition, transaction of set(Adding, Union, intersection), working with sets	

Unit IV	**Modules ,Working with files, Exception handling**	**6 Hours**
	1. Modules: Importing module, Creating & exploring modules, Math module, Random module, Time module 2. Packages: Importing package, creating package, examples 3. Working with files: Creating files and Operations on files (open, close, read, write), File object attributes, file positions, Listing Files in a Directory, Testing File Types, Removing files and directories, copying and renaming files, splitting pathnames, creating and moving directories 4. Regular Expression- Concept of regular expression, various types of regular expressions, using match function. Exception Handling: Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions.	

Suggested Readings	
1.	An Introduction to Computer Science using Python by Jason Montojo, Jennifer Campbell, PaulGries, The pragmatic bookshelf-2013
2.	James Payne, "Beginning Python: Using Python and Python 3.1, Wrox Publication
3.	Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication Learning with Python ",Green Tea Press, 2002
4.	Introduction to Problem Solving with Python by E balguruswamy,TMH publication- 2016
5.	Beginning Programming with Python for Dummies Paperback – 2015 by John Paul Mueller
6.	Object-oriented Programming in Python, Michael H. Goldwasser, David Letscher,Pearson Prentice Hall-2008

Website Reference Links	
1.	https://www.programiz.com/python-programming
2.	https://www.w3schools.com/python/python_intro.asp
3.	https://www.geeksforgeeks.org/python-programming-language
4.	https://www.analyticsvidhya.com/

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Software Engineering**
Course Code	**23SBCS33MM**
Semester	**III**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	To get knowledge and understanding of software engineering discipline.
2.	To learn analysis and design principles for software project development.
3.	To Understand Object Oriented Modeling techniques and their applicability.

Course Outcome	
1.	Compare and choose a process model for a software project development.
2.	Identify requirements analyze and prepare models
3.	Prepare the SRS, Design document, Project plan of a given software system
4.	To learn and identify the components of Unified Modeling Language

Syllabus		
Unit I	**Introduction To Software Engineering and Process Models**	**6 Hours**
	1. Definition of Software 2. Nature of Software Engineering 3. Changing nature of software 4. Software Process 4.1.The Process Framework 4.2.Umbrella Activities 4.3.Process Adaptation 5. Generic Process Model 6. Prescriptive Process Models 7. The Classical Waterfall Model 8. Incremental Process Models 9. Evolutionary Process Models 10. Requirement Analysis 11. Elicitation 12. Software Requirement Specification (SRS)	
Unit II	**Agile Development**	**5 Hours**
	1. What is Agility? 2. Agile Process 2.1 Agility Principles 2.2 The Politics Of Agile Development 2.3 Human Factors 3. Extreme Programming(XP) 3.1 XP Values 3.2 XP Process 3.3 Industrial XP 4. Definition of Scrum 4.1 Scrum Life Cycle	
Unit III	**Structural Modeling**	**6 Hours**
	1. Concept of UML 2. Advantages of UML 3. Classes 4. Relationship 4. Interface 5. Types and Roles 6. Packages 7. Class Diagram 8. Object Diagram	
Unit IV	**Behavioral modeling**	**7 Hours**
	1. Interactions 2. Use Cases and Use Case Diagram with stereotypes 3. Interaction Diagram 4. Sequence Diagram 4. Activity Diagram 5. State Chart Diagram	

Unit V	Architectural modeling	6 Hours
	1. Component 2. Components Diagram 3. Deployment Diagram 4. Collaboration Diagram	

Suggested Readings	
1.	Software Engineering : A Practitioner's Approach - Roger S. Pressman, McGrawhill(Eighth Edition) ISBN-13: 978-0-07-802212-8, ISBN-10: 0-07- 802212-6
2.	A Concise Introduction to Software Engineering - Pankaj Jalote, Springer ISBN: 978- 1-84800-301-9
3.	The Unified Modeling Language Reference Manual - James Rambaugh, Ivar Jacobson,Grady Booch ISBN 0-201-30998-X

Website Reference Links	
1.	https://www.springboard.com/blog/software-engineering
2.	https://www.geeksforgeeks.org/software-engineering-agile-software-development/
3.	https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml
4.	https://www.javatpoint.com/uml-diagrams

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Lab course in Python Programming**
Course Code	**23SBCS34MM**
Semester	**III**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	To introduce programming concepts using python.
2.	To develop Programming logic using python.
3.	To develop basic concepts and terminology of python programming
4.	To test and execute python programs

Course Outcome	
1.	Students should be able to develop logic for problem solving
2.	Students should be able to determine the methods to create and develop Python programs by utilizing the data
3.	Student should understand the structures like lists, dictionaries, tuples and sets.
4.	Students should be able to write python programs and develop a small application project

Assignment No.	Assignments	No. of Sessions
1.	Assignment No. 1 based on Python Basics and IDE, Simple Python Programs	01
2.	Assignment No. 2 based Strings and Functions	02
3.	Assignment No. 3 based on List	01
4.	Assignment No. 4 based on Tuples	01
5.	Assignment No. 5 based on Dictionary	02
6.	Assignment No. 6 based on Sets	01
7.	Assignment No. 7 based on File Handling and Date-Time	01
8.	Assignment No. 8 based on Exception handling and Regular expression	01

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Lab Course in Data Structures and Algorithms – I**
Course Code	**23SBCS31VS**
Semester	**III**
No. of Credits	**2**
Course Type	**VSC**

Course Objectives	
1.	To learn the systematic way of solving problem.
2.	To understand the different methods of organizing large amount of data.
3.	To efficiently implement the different data structures.
4.	To efficiently implement solutions for specific problems.
5.	To apply linear data structures.

Course Outcome	
1.	To use well-organized data structures in solving various problems.
2.	To differentiate the usage of various structures in problem solution.
3.	Implementing algorithms to solve problems using appropriate data structures.

Assignment No.	Assignments	No. of Sessions
1.	Assignment No. 1 based on Searching Algorithms	01
2.	Assignment No. 2 based on Sorting Algorithms	02
3.	Assignment No. 3 based on Singly Linked List	02
4.	Assignment No. 4 based on Doubly Linked List	02
5.	Assignment No. 5 based Stack	01
6.	Assignment No. 6 based Queue	02

B.Sc. Computer Science
Semester IV

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Data Structures and Algorithms – II**
Course Code	**23SBCS41MM**
Semester	**IV**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	To learn the systematic way of solving problem.
2.	To design algorithms.
3.	To understand the different methods of organizing large amount of data
4.	To efficiently implement solutions for specific problems.
5.	To efficiently implement the non-linear data structures.

Course Outcome	
1.	Implementation of different data structures efficiently.
2.	Usage of well-organized data structures to handle large amount of data.
3.	Usage of appropriate data structures for problem solving.

Syllabus		
Unit I	**Tree**	**10 Hours**
	1. Concept and Terminologies 2. Types of Binary trees - Binary tree, skewed tree, strictly binary tree, full binary tree, complete binary tree, expression tree, binary search tree, Heap 3. Representation – Static and Dynamic 4. Implementation and Operations on Binary Search Tree - Create, Insert, Delete, Search, Tree traversals– preorder, inorder, postorder (recursive implementation),Counting leaf, non-leaf and total nodes, Copy, Mirror. 5. Applications of trees 5.1 Heap sort implementation	
Unit II	**Efficient Search Trees**	**6 Hours**
	1. Terminology: Balanced trees - AVL Trees 2. AVL Tree- concept and rotations 3. Multi-way search tree - B and B+ tree - Insertion, Deletion	
Unit III	**Graph**	**8 Hours**
	1. Concept and terminologies 2. Graph Representation –Adjacency matrix, Adjacency list, Inverse Adjacency list, Adjacency multilist 3. Graph Traversals – Breadth First Search and Depth First Search (with implementation) 4. Applications of graph 4.1 Topological sorting 4.2 Use of Greedy Strategy in Minimal Spanning Trees (Prims and Kruskals algorithm) 4.3 Single source shortest path - Dijkstra's algorithm 4.4 Dynamic programming strategy, All pairs shortest path - Floyd Warshall algorithm 4.5 Use of graphs in social networks	
Unit IV	**Hash Table**	**6 Hours**
	1. Concept of hashing 2. Terminologies – Hash table,Hash function, Bucket, Hash address, collision, synonym,overflow etc. 3. Properties of good hash function 4. Hash functions : division function, MID square , folding methods 5. Collision resolution techniques 5.1 Open Addressing - Linear probing, quadratic probing, rehashing 5.2 Chaining - Coalesced , separate chaining	

Suggested Readings	
1.	Fundamentals of Data Structures in C- Ellis Horowitz, SartajSahni,Susan Anderson-Freed, 2nd Edition, Universities Press.
2.	Data Structures using C and C++-YedidyahLangsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Pearson Education
3.	Data Structures: A Pseudo code approach with C, Richard Gilberg ,Behrouz A. Forouzan, Cengage Learning.
4.	Introduction to Data Structures in C-Ashok Kamthane, Pearson Education
5.	Algorithms and Data Structures, Niklaus Wirth, Pearson Education
6.	6. Introduction to Algorithms—Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein--MIT Press

Website Reference Links	
1.	https://www.javatpoint.com/what-is-a-non-linear-data-structure
2.	https://ace.nus.edu.sg/course/data-structure-and-algorithm-part-ii/
3.	https://www.geeksforgeeks.org/data-structures/

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc. (Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Object Oriented Programming Using C++**
Course Code	**23SBCS42MM**
Semester	**IV**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	Learn how to write code in a way that it is independent of any particular type
2.	Understanding the process of exposing the essential data to the outside of the world and hiding the low level data
3.	Be able to explain the difference between object oriented programming and procedural programming.
4.	Understanding the concept of data abstraction and encapsulation , how to Design c++ classes for code reuse, how to implement copy constructors and class member functions, to overload functions and operators in c++.

Course Outcome	
1.	Identify importance of object oriented programming and difference between structured oriented and object oriented programming features.
2.	Able to make use of objects and classes for developing programs.
3.	Implement programming techniques to solve problems in the C++ programming language..

Syllabus		
Unit I	**Introduction to Object Oriented Concepts**	**4 Hours**
	1. Concept of Object Oriented Programming, 2. Object oriented programming vs. procedure programming 3. Characteristics of Object Oriented Programming 3.1 Classes 3.2 Object 3.3 Abstraction 3.4 Inheritance 3.5 Polymorphism 3.6 Data Binding 3.7 Encapsulation	
Unit II	**Programming Basics**	**8 Hours**
	1. C++ basic structure 2. Simple "Hello World" program 3. Compiling, linking and running a C++program 4. Managing Console I/O 5. Data Types and type conversion 6. New operators and keywords 7. Type casting in C++ 8. Reference variables, 9. Usage of namespace, 10. Usage of Manipulators 10.1 Endl 10.2 Ws 10.3 Ends 10.4 Flush 10.5 setw(val)	
Unit III	**Classes and Objects**	**6 Hours**
	1. Introduction to Class and object 2. Defining data members and member functions 3. Access Specifies of class 4. Static data members & Static member functions 5. Arrays and Array of objects 6. Objects as function arguments	
Unit IV	**Functions**	**6 Hours**
	1. Passing argument and returning values from function 2. Call and return by reference 3. Inline functions 4. Default and Const function arguments 5. Friend Function 6. Function overloading 7. Constructors 7.1 Default constructor 7.2 Parameterized constructor 7.3 Copy constructor 7.4 Multiple constructors 7.5 Constructors with default arguments	

	7.6 Dynamic constructor 8. Destructors 9. Operator overloading	
Unit V	**Inheritance**	**6 Hours**
	1. Introduction to Inheritance 2. Defining derived classes & Visibility modes 3. Types of inheritance 3.1 Single 3.2 Multilevel 3.3 Multiple 3.4 Hierarchical 3.5 Hybrid inheritance 4. Constructors and destructors in derived classes 5. Virtual base classes and Abstract classes	

Suggested Readings	
1.	E Balagurusamy, "Object Oriented Programming with C++", Fifth edition 2011, Tata McGraw-Hill
2.	Robert Lafore, "Object Oriented Programming in Turbo C++" First Edition, Galgotia Publications.
3.	Herbert Schildt, "Compete Reference C++", 4th Edition 2003, McGraw-Hill Publication
4.	Ashok N. Kamthane, Object oriented Programming with ANSI & Turbo C++, First Edition, Pearson India
5.	D Ravichandran, Programming with C++, Second edition, Tata McGraw-Hil

Website Reference Links	
1.	https://www.w3schools.com/cpp/
2.	https://onecompiler.com/cpp
3.	https://www.javatpoint.com/cpp-tutorial
4.	https://www.scaler.com/topics/cpp/

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Evolution of Network and Data Communication**
Course Code	**23SBCS43MM**
Semester	**IV**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	To give clear idea of signals, in data communications and their correction, networks classes and standard organization, etc.
2.	To prepare students with basic networking concepts: data communication, protocols and standards, various topologies and applications of network
3.	To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model
4.	To understand different protocols of application layer.

Course Outcome	
1.	To have a good understanding of the OSI and TCP/IP Reference Models and in particular have a good knowledge of Layer.
2.	Describe the functions of each layer in ISO/OSI model.
3.	To understand the working of various protocols.
4.	Student will understand the different protocols of Application layer.
5.	Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies

Syllabus		
Unit I	**Evolution Of Computer Networks**	**6 Hours**
	1. What is Computer Networking? 2. Timeline of Computer Networks 3. Root of Computer Networks 4. Computer networks as a Result of the computing and communications Technology 5. Batch-Processing Systems 6. Evolution of Networking 6.1 Ancient Networks 6.2 Postal Networks 6.3 Telephone Networks 7. History of Computer Networking 8. What is ARPANET - the First Network 8.1 Internet 8.2 Telnet 8.3 World Wide Web 9.Communication Technologies – Terminologies	
Unit II	**Introduction to Data Communication and Networking**	**8 Hours**
	1. Why study data communication? 2. Data Communication, Networks, Protocols and Standards, Standards Organizations. 3. Elements of Data Communication 3.1 Data Representation 3.2 Data Flow 4. Networks, network criteria, network types - LAN, WAN, Switching, The Internet, Accessing the Internet 5. Network Software- Protocol hierarchies, Design Issues of the layer, Connection Oriented and Connectionless Services, 6. Reference models - OSI Reference Models, TCP/IP Reference model, Connection devices in different layers, Comparison of OSI and TCP/IP Reference Models.	

Unit III	**Network Layer**	**5 Hours**
	1. Network layer services - Packetizing, Routing and forwarding, other services 2. Open and closed loop congestion control 3. IPv4 addressing- Address space, classful addressing, Subnetting, Supernetting, classless addressing, Network address 4. Forwarding of IP packets- based on destination address, based on label 5. Network Layer Protocols- Internet Protocol (IP), IPv4 datagram format, Fragmentation, packet format, extension header, Difference between IPv4 and IPv6	
Unit IV	**Transport Layer**	**6 Hours**
	1. Transport layer Services- Process-to-process communication, Addressing, Encapsulation and decapsulation, Multiplexing and demultiplexing, Flow control, Pushing or pulling, Flow control, Buffers, Sequence numbers. 2. Connectionless and Connection-oriented service, Port numbers. 3. Transport layer protocols- User datagram protocol, user datagram, UDP services. 4. Transmission Control Protocol - TCP Services, TCP Features, TCP Segment format, three-way handshake for connection establishment and termination.	
Unit V	**Application Layer**	**5 Hours**
	1. Domain Name System 1.1. Name space-Flat name space, Hierarchical name space 1.2. Domain Name Space -Label ,Domain name, FQDN,PQDN 1.3. Distribution of Domain Name Space. 1.4. DNS in the Internet: Generic domains, Country domains, inverse domain. 1.5. Resolution-Resolver, mapping names to address, mapping addresses to names, recursive resolution, Iterative resolution, caching 2. Electronic Mail- Architecture-First scenario, second scenario, Third scenario, Fourth scenario User agent-services of user agent, types of UA Format of e-mail MIME-MIME header Message transfer agent-SMTP Message Access Agent: POP and IMAP 4 File Transfer FTP-Communication over data control connection, File type, data structure, Transmission mode, anonymous FTP	

Suggested Readings	
1.	Data communication & Networking by Bahrouz Forouzan.
2.	Computer Networks by Andrew S. Tanenbaum
3.	Data and Computer Communications by William Stallings
4.	Andrew s. Tanenbaum .Computer Networks. Pearson Education .Fourth Edition.

Website Reference Links	
1.	https://www.geeksforgeeks.org/difference-between-lan-man-and-wan/
2.	https://www.tutorialspoint.com/data_communication_computer_network/index.htm
3.	https://www.techtarget.com/searchnetworking/definition/Network-layer
4.	https://www.imperva.com/learn/application-security/osi-model/

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc. (Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Lab Course in Object Oriented Programming Using C++**
Course Code	**23SBCS44MM**
Semester	**IV**
No. of Credits	**2**
Course Type	**Major**

Course Objectives	
1.	To learn the syntax and semantics of the C++ Programming language
2.	To learn the object oriented programming paradigm and use of classes along with the fundamental of object oriented design.
3.	To learn how inheritance and virtual functions.

Course Outcome	
1.	Design object oriented solutions for small systems involving multiple objects
2.	Apply object oriented software principles in problem solving
3.	Develop the application using object oriented programming language.

Assignment No.	Assignments	No. of Sessions
1.	Assignment No. 1 based on basic cpp programs	01
2.	Assignment No. 2 based on Classes and objects	02
3.	Assignment No. 3 based on Functions	02
4.	Assignment No. 4 based on constructors , destructors and operator overloading	02
5.	Assignment No. 5 based on inheritance	03

M. C. E. Society's
Abeda Inamdar Senior College
Of Arts, Science and Commerce, Camp, Pune-1 (Autonomous)
Affiliated to Savitribai Phule Pune University NAAC accredited 'A' Grade

S.Y.B.Sc.(Computer Science) 2024-25

(CBCS – Autonomy 23 Pattern)

Course/ Paper Title	**Lab Course in Data Structures and Algorithms – II**
Course Code	**23SBCS41SE**
Semester	**IV**
No. of Credits	**2**
Course Type	**SEC**

Course Objectives	
1.	To learn the systematic way of solving problem.
2.	To design algorithms.
3.	To understand the different methods of organizing large amount of data
4.	To efficiently implement solutions for specific problems.
5.	To efficiently implement the non-linear data structures.

Course Outcome	
1.	Implementation of different data structures efficiently.
2.	Usage of well-organized data structures to handle large amount of data.
3.	Usage of appropriate data structures for problem solving.

Assignment No.	Assignments	No. of Sessions
1.	Assignment No. 1 based on Binary Search Tree and Traversals	02
2.	Assignment No. 2 based on Binary Search Tree Operations	02
3.	Assignment No. 3 based on Graph implementation	02
4.	Assignment No. 4 based on Graph Applications	02
5.	Assignment No. 5 based on Hash Table	02