Deccan Education Society's
FERGUSSON COLLEGE (AUTONOMOUS), PUNE

Syllabus

for

S. Y. B. Sc. (Computer Science)

[Pattern 2019]

(B.Sc. Semester-III and Semester-IV)

From Academic Year

2020-2021

Deccan Education Society's
Fergusson College (Autonomous), Pune

S.Y. B.Sc. ComputerScience(Pattern 2019)

From academic year2020-21

Particulars	Name of Paper	Paper Code	Title of Paper	No. of Credits
S.Y. B.Sc. Semester III	Theory Paper - 1	CSC2301	Data Structures and Algorithms – I	2
	Theory Paper - 2	CSC2302	Software Engineering	2
	Practical Paper - 1	CSC2303	Computer Science Practical -III	2
S.Y. B.Sc. Semester IV	Theory Paper - 3	CSC2401	Data Structures and Algorithms – II	2
	Theory Paper - 4	CSC2402	Computer Networks - I	2
	Practical Paper - 2	CSC2403	Computer Science Practical -IV	2

S.Y. B.Sc. Semester III		
Title of the Course and Course Code	**Data Structures and Algorithms – I (CSC2301)**	**Number of Credits : 02**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Identify and define appropriate algorithms by developing problem solving skills by analysing a problem.	
CO2	Discuss about different data structures and compare between primitive and non-primitive data structures.	
CO3	Apply the various algorithms to solve real world computing problems.	
CO4	Analyze the algorithms on the scale of their performance.	
CO5	Test and perform critical evaluation of the program outcome to validate the correctness of the algorithm.	
CO6	Integrate various concepts of algorithmic solutions and develop effective algorithms. Design different algorithms and compare their performance.	

UnitNo.	Title of Unit and Contents	No of Lectures
I	**Introduction to Data Structure and algorithm analysis** Concept 1.1 Data Type, Data Object, Abstract Data Type(ADT) 1.2 Need, Types of Data Structure 1.3 Applications of Data Structure 1.4 Algorithm types 1.5 Algorithm Analysis : Complexity (Time, Space),Asymptotic Notations (big O notation, Omega, Theta)	**3**
II	**Linear Data Structure - Array** 2.1 Array as ADT 2.2 Representation 2.3 Applications 2.3.1 Sorting: Concept, terminology, types. Methods: Bubble Sort, Insertion Sort, Selection sort, Quick Sort, Merge Sort, Radix sort. Comparison of sorting methods. 2.3.2 Searching: Linear,Binary	**8**
III	**Linear Data Structure - Stack** 3.1 Introduction 3.2 Static representation of Stack Operations (Init, Push, Pop, Peek) 3.3 Recursion 3.4 Applications: String reversal, Parenthesis balancing, polish notation, Evaluation, Backtracking, 4 queens' problem	**6**
IV	**Linear Data Structure- Queue** 4.1 Introduction 4.2 Representation of queue: Static	**7**

	4.3 Operations(Insert, Delete, Display) 4.4 Types of queue (Circular, Priority, Dequeue) 4.5 Applications : Job scheduling with priority	
V	**Linear Data Structure- Linked List** 5.1 Introduction, types (singly, doubly, circular) 5.2 Representation (dynamic) 5.3 Operations on linked list (Create, insert, delete, Search, traverse) 5.4 Dynamic implementation of stack and queue using singly linked list 5.5 Generalized linked list (Concept, representation, Example) 5.6 Applications:Polynomial manipulation, memory management in OS(FIFO)	**12**

References:

1. Ellis *Horowitz*, Sartaj*Sahni* and SanguthevarRajasekara, Fundamentals of Computer Algorithms, Galgotia Pub. 2001 ed.
2. Y. Langsam, M. AugensteinAnd A. M. Tenenbaum, Data Structures using C & C++, Prenctice-Hall International.
3. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill.
4. Gilberg&Forouzan, Data Structures: A Pseudo-code approach with C, Thomson Learning.
5. S.K.Srivastava and Deepali Srivastava, Data Structures Through C in Depth, BPB Publication

Title of the Course and Course Code	Software Engineering (CSC2302)	Number of Credits : 02
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Define software and state its characteristics and features.	
CO2	Illustrate the concepts of Object oriented approach towards software development.	
CO3	Design and apply various software models to solve real world use cases of software projects.	
CO4	Analyze the UML diagrams on the scale of their application in software development and classify various architectural designs of the project using various UML diagrams.	
CO5	Test and perform critical evaluation of the software models and diagrams by applying them to various use cases of real world problems.	
CO6	Integrate the concepts of developing software projects to meet the needs of the software industry.	

Unit No.	Title of Unit and Contents	No of Lectures
I	**Introduction to Software Engineering and Process Models** 1.1 Definition of Software 1.2 Nature of Software Engineering 1.3 Feasibility Study of General System 1.4 Object oriented concepts 1.5 OOSDLC Model 1.6 Prescriptive Process Models 1.6.1 The Waterfall Model 1.6.2 Incremental Process Models 1.6.3 Evolutionary Process Models 1.6.4 Concurrent Models 1.7 Agile Model	6
II	**Requirements Analysis and UML** 2.1 Requirement Engineering Tasks 2.1.1 Inception 2.1.2 Elicitation 2.1.3 Elaboration 2.1.4 Negotiation 2.1.5 Specification 2.1.6 Validation 2.2 Software requirement specification (SRS) 2.3 Unified Modelling Language 2.3.1 Advantages and Features of UML 2.3.2 UML Diagrams	4

III	**UML for Database Oriented Project** 3.1 Selection of topic and Scenario Discussion 3.2 Defining Objects and Classes 3.3 Requirement Analysis 3.4 Actors and types of actors 3.5 Use-case Diagram 3.6 Class Based Model 3.6.1 Difference between class element and object element 3.6.2 Relationships 3.6.3 Types of relationships 3.6.4 Class diagram 3.6.5 Object diagram 3.7 Behavioural Model 3.7.1 Sequence diagram 3.7.2 Activity diagram 3.8 Package Diagram 3.9 Architectural Design Model 3.9.1 Component diagram 3.9.2 Deployment diagram	**13**
IV	**UML for Game and Utility Based Project** 4.1 Selection of topic and Scenario Discussion 4.2 Use-case Diagram 4.3 Class Diagram 4.4 Behavioural Model 4.4.1 Collaboration diagram 4.4.2 State Diagram 4.5 Architectural Design Model 4.5.1 Component diagram [optional for the game project] 4.5.2 Deployment diagram	**8**
V	**Software Testing** 5.1 What is testing? 5.2 Different Testing Techniques 5.3 Writing Test cases 5.4 Validation testing	**5**

References:

1. Grady Booch, James Rambaugh, The Unified Modeling Language User / Reference Guide, Pearson Education INC
2. Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC
3. Craig Larman, Applying UML and Patterns, Pearson Education INC
4. Bennett, Simon, Object Oriented Analysis and Design, McGraw Hill
5. Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill (Eighth Edition) ISBN-13: 978-0-07-802212-8, ISBN-10: 0-07-802212-6

Title of the Course and Course Code	Computer Science Practical - III (CSC2303)	Number of Credits : 02
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Describe various concepts of different types of data structures. Define the concepts of effective software engineering practices, software modelling processes and Unified Modelling Language (UML).	
CO2	Explain various techniques of data structures to give efficient solutions to the computing problems. Discuss software engineering principles and UML components.	
CO3	Apply various searching, sorting algorithms to develop programs. Use software engineering practices for modelling of real word use cases using UML components.	
CO4	Analyze and compare different searching, sorting techniques and different data structures to arrange data in memory. Classify different software modelling processes to decide the best process for the given requirements.	
CO5	Test and perform efficient programs using various data structures. Evaluate real word use cases using software engineering principles and UML.	
CO6	Write efficient programs by integrating various searching, sorting techniques and data structures. Integrate various components of UML to model any real word use case.	

List of Practicals (Compulsory 10 + 2 Activity)

Sr. No.	Title of Experiment / Practical
1	Searching Algorithms
2	Sorting Algorithms
3	Stack
4	Queue
5	Linked List
6	Activity
7	Problem Definition, Feasibility Study and Software Requirement Specification
8	Use Case Diagram
9	Class diagram
10	Behavioral Model (Sequence, Activity, State Diagram)
11	Architectural Model (Collaboration, Component, Deployment Diagram)

12	UML Case Study (Activity)

S.Y. B.Sc. Semester IV		
Title of the Course and Course Code	**Data Structures and Algorithms – II (CSC2401)**	**Number of Credits : 02**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Study different algorithms of non-linear data structures.	
CO2	Illustrate various concepts for developing algorithmic solutions.	
CO3	Apply various algorithms to solve real world computing problems.	
CO4	Analyze the algorithms on the scale of their performance.	
CO5	Test and implement critical evaluation of the program outcome to validate the correctness of the algorithm.	
CO6	Develop problem solving skills by analysing a problem and by identifying and defining appropriate algorithms.	

Unit No.	Title of Unit and Contents	No of Lectures
I	**Non-Linear Data Structure - Tree** 1.1 Concept and terminologies 1.2 Binary Search Tree (BST) 1.3 Representation (Static, dynamic) 1.4 Operations on BST (Create, insert, delete) 1.5 Traversals (inorder, preorder, postorder), counting of nodes, 1.6 Application: Heap Sort, AVL tree, Huffman Coding, Expression tree 1.7 AVL Trees, Red Black Trees, M-way search tree, B and B+ tree	**18**
II	**Non-Linear Data Structure - Graph** 2.1 Concept and terminologies, 2.2 Representation (Adjacency matrix, Adjacency list, Adjacency Multilist) 2.3 Traversal (BFS, DFS) 2.4 Applications (Shortest path algorithm: Dijkstra's algorithm, Bellman ford, Floyd Warshalls) 2.5 AOV Network – Concept 2.6 AOE Network – Concept 2.7 Prims and Kruskal's algorithm	**12**
III	**Hashing** 3.1 Terminologies 3.2 Properties of good hash function 3.3 Hash Functions: Division Function, MID Square, Folding methods.	**6**

	3.4 Collision resolution techniques: Open addressing: Linear Books: Quadratic probing, Rehashing. Chaining: Coalesced, separate chaining	

References:

1. Ellis *Horowitz*, Sartaj*Sahni* and SanguthevarRajasekara, Fundamentals of Computer Algorithms, Galgotia Pub. 2001 ed.
2. Y. Langsam, M. Augensteinand A. M. Tenenbaum, Data Structures using C & C++, Prenctice-Hall International.
3. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, Tata McGraw Hill.
4. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, Thomson Learning.
5. S. K. Srivastava and Deepali Srivastava, Data Structures Through C in Depth, BPB Publication

S.Y. B.Sc. Semester IV		
Title of the Course and Course Code	**Computer Networks – I (CSC2402)**	**Number of Credits : 02**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Define basic concepts of computer networks.	
CO2	Explain networking models and addressing schemes.	
CO3	Apply the topologies, encoding schemes and addressing methods as per given network environments.	
CO4	Analyze the working of various layers of network models.	
CO5	Compare various topologies, networking types and protocols.	
CO6	Integrate the concepts of computer networking to design set up of network environments.	

Unit No.	Title of Unit and Contents	No of Lectures
I	**Introduction to Computer Network** 1.1 Data Communication - Definition, components, data representation, Data Flow, Key elements of protocol, Standards, Standards organizations 1.2 Network Hardware - Broadcast and point-to-point 1.3 Network Types-LAN, MAN, WAN, Wireless Networks, Home Networks, Internetwork 1.4 Topologies - bus, star, ring, mesh, hybrid 1.5 Design issues of the layers - addressing, error control, flow control, multiplexing and demultiplexing, routing, Connection-oriented andconnectionless service	**4**
II	**Network Models** 2.1 OSI Reference Model - Functionality of each layer	**4**

	2.2 TCP/IP Reference Model - Functionality of each layer 2.3 Comparison of OSI and TCP / IP Model 2.4 Addressing - Physical, Logical and Port addresses	
III	**The Physical Layer** 3.1 Transmission impairment - Attenuation, Distortion and Noise 3.2 Line Coding Characteristics, Line Coding Schemes - Unipolar - NRZ, Polar-NRZ-I, NRZ-L, RZ, Manchester and Differential Manchester, AMI, HDB3, B8ZS 3.3 Switching - Circuit Switching, Message Switching and Packet Switching 3.4 Comparison of circuit & packet switching 3.5 Application of switching - Telephone and mobile network 3.6 Physical Layer Device - Repeater	**8**
IV	**The Data Link Layer** 4.1 Design Issues - Services provided to the Network Layer 4.2 Framing - Concept, methods - Character Count, Flag bytes with Byte Stuffing, Starting & ending Flags with Bit Stuffing and Physical Layer Coding Violations, Error Control, Flow Control 4.3 Data Link Layer Protocols –Noiseless channel -A Simplex, Stop-And-Wait protocol, Noisy channel –stop & wait, ARR, Pipelining, Go –back –N, ARR & ARQ, Concept of sliding window, selective repeat ARR 4.4 Piggybacking-Need, Advantages / Disadvantages 4.5 Data Link Layer Device – Bridge	**10**
V	**Multiple Access** 5.1 Random Access Protocols ALOHA - pure and slotted Controlled Access Protocol - Reservation, Polling and Token Passing	**2**
VI	**Network Layer** 6.1 Design Issues of network layer - Store and forward packet switching, services provided to transport layer, Implementation of connectionless and connection oriented services, comparison of virtual circuit and datagram 6.2 Logical Addressing - IPV4, addresses, Address space, Notations, Classful addressing, Classless addresses 6.3 IPV4 Datagram format 6.4 Network layer device - Router, Router table 6.5 Network addresstranslation	**8**

References:

1. Andrew Tanenbaum, Computer Networks, Pearson Education. [4th Edition]
2. Behrouz Forouzan, Data Communication and Networking, Tata McGraw Hill. [4th Edition]

S.Y. B.Sc. Semester IV		
Title of the Course and Course Code	**Computer Science Practical – IV (CSC2403)**	**Number of Credits : 02**
Course Outcomes (COs) On completion of the course, the students will be able to:		
CO1	List the errors and warnings of programs based on non-linear data structures.	
CO2	Represent various data structures for trees, graphs and hashing techniques.	
CO3	Implement various shell interface commands by constructing and studying the Linux Environment.	
CO4	Analyze the network behavior using the Wireshark tool.	
CO5	Determine the implementation of user management on Linux based web browsers and different types of servers.	
CO6	Write the program to simulate behavior of various non-linear data structures.	

List of practicals (Compulsory 10 + 2 Activity)

Sr No.	Title of Experiment / Practical
1	Assignments on Trees - I
2	Assignments on Trees - II
3	Assignments on Graphs - I
4	Assignments on Graphs - II
5	Assignments on Hashing methods
6	Activity
7	Study of Linux environment
8	Study of Shell Interface
9	Network based Linux commands
10	User management in Linux
11	Study of various web browsers and types of servers used in Linux
12	Case study: Wireshark Tool (Activity)