Fergusson College (Autonomous), Pune

Learning Outcomes-Based Curriculum

for 3/4 years B. Sc. /B. Sc. (Honours) Programme

as per guidelines of

NEP-2020

for

S.Y B. Sc. (Computer Science)

With effect from Academic Year

2024-2025

Semester	Paper	Paper Code	Paper Title	Type	Credits
III	MAJOR	CSC-201	Data Structures	Theory	4
		CSC-200	Computer Science Practical -3	Practical	2
	OE-5	CSC-220	Basics of Data Analytics	Theory	2
	VSC	CSC-230	Web Page Designing	Theory	2
	SEC	CSC-240	Software Engineering Approach Using UML	Skill	2
	CEP	CEP-245	Foundations of Community Engagement	CEP	2
	MINOR	MTS - 215	Applied Linear Algebra	Theory	2
	MINOR	MTS - 216	Mathematics Practical - 3	Practical	2
	OR				
	MINOR	ELS - 215	Embedded Systems	Theory	2
	MINOR	ELS - 216	Electronics Practical - 3	Practical	2
	OR				
	MINOR	STS – 215	Statistical Methods I	Theory	2
	MINOR	STS - 216	Statistics Practical - 3	Practical	2
IV	MAJOR	CSC - 251	Object Oriented Programming using C++	Theory	4
		CSC - 250	Computer Science Practical – 4	Practical	2
	OE-6	CSC - 270	e-Commerce	Theory	2
	VSC	CSC -280	Advanced Web Page Designing	Theory	2
	SEC	CSC - 290	Fundamentals of Software Testing	Skill	2
	FP	FP - 295	Community Engagement - Field Project	Field Project	2
	MINOR	MTS - 265	Computational Geometry	Theory	2
	MINOR	MTS - 266	Mathematics Practical - 4	Practical	2
	OR				
	MINOR	ELS - 265	Fundamentals of Internet of Things (IoT)	Theory	2
	MINOR	ELS - 266	Electronics Practical - 4	Practical	2
	OR				
	MINOR	STS - 265	Statistical Methods II	Theory	2
	MINOR	STS - 266	Statistics Practical - 4	Practical	2

Teaching and Evaluation (Only for FORMAL education courses)

Course Credits	No. of Hours per Semester Theory/Practical	No. of Hours per Week Theory/Practical	Maximum Marks	CE 40 %	ESE 60%
1	15 / 30	1 / 2	25	10	15
2	30 / 60	2 / 4	50	20	30
3	45 / 90	3 / 6	75	30	45
4	60 / 120	4 / 8	100	40	60

S.Y.B.Sc Computer Science (Semester III)		
CSC-201	**Data structures (Major Theory)**	**Credits : 04 Hours : 60**
	Course Outcomes (COs) On completion of the course, the students will be able to:	**Bloom's Cognitive Level**
CO1	Identify and define appropriate algorithms by developing problem solving Skills by analyzing a problem	1
CO2	Illustrate various concepts for developing algorithmic solutions.	2
CO3	Apply various algorithms to solve real world computing problems.	3
CO4	Analyze the algorithms on the scale of their performance.	4
CO5	Test and perform critical evaluation of the program outcome to validate the Correctness of the algorithm.	5
CO6	Integrate various concepts of algorithmic solutions and develop effective algorithms. Design different algorithms and compare their performance.	6

Unit No	Contents	No. of Hours
I	**Structures ,Unions and File handling** 1.1 Structures and Unions 1.1.1 Creating structures 1.1.2 Structure declaration and initialization 1.1.3 Accessing structure members (dot Operator) 1.1.4 Array of structures 1.1.5 Passing structures to functions 1.1.6 Nested structures 1.1.7 Pointers and structures 1.1.8 Self-referential structure 1.1.9 Unions: Declaration, Initialization and accessing 1.1.10 Difference between structures and unions 1.1.11 Typedef 1.2 File Handling 1.2.1 Introduction – streams 1.2.2 Types of files 1.2.3 Modes of file opening 1.2.4 Operations on files 1.2.5 Random access to files	**06**
II	**Introduction to Data Structure and algorithm analysis concept** 2.1 Structure , Union, File handling 2.2 Data Type, Data Object, Abstract Data Type(ADT)	**03**

	2.3 Need, Types of Data Structure 2.4 Applications of Data Structure 2.5 Algorithm types 2.6 Algorithm Analysis : Complexity (Time, space), Asymptotic Notations (big O notation, Omega, Theta)	
III	**Linear Data Structure - Array** 3.1 Array as ADT 3.2 Representation 3.3 Applications 3.3.1 Sorting: Concept, terminology, types. Methods: Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Radix sort. Comparison of sorting methods. 3.3.2 Searching: Linear, Binary	**07**
IV	**Linear Data Structure - Stack** 4.1 Introduction 4.2 Static representation of Stack Operations (Init, Push, Pop, Peek) Recursion 4.3 Applications: String reversal, Parenthesis balancing, polish notation, Evaluation	**05**
V	**Linear Data Structure- Queue** 5.1 Introduction 5.2 Representation of queue: Static 5.3 Operations(Insert, Delete, Display) 5.4 Types of queue (Circular, Priority, Dequeue) 5.5 Applications : scheduling in Operating system	**05**
VI	**Linear Data Structure- Linked List** 6.1 Introduction, types (singly, doubly, circular) 6.2 Representation (dynamic) 6.3 Operations on linked list (Create, insert, delete, Search, traverse) 6.4 Dynamic implementation of stack and queue using singly linked list 6.5 Generalized linked list (Concept, representation, Example) 6.6 Applications: Polynomial manipulation 6.7 Case studies on Linear data structures	**10**
VII	**Non-Linear Data Structure – Tree** 7.1 Concept and terminologies 7.2 Binary Search Tree (BST) 7.3 Representation (Static, dynamic) 7.4 Operations on BST (Create, insert, delete) 7.5 Traversals (inorder, preorder, postorder), counting of nodes, 7.6 Application: Heap Sort, AVL tree, Expression tree	**10**
VIII	**Non-Linear Data Structure – Graph** 8.1 Concept and terminologies, Representation (Adjacency matrix, Adjacency list, Adjacency Multilist 8.2 Traversal (BFS, DFS) 8.3 Applications (Shortest path algorithm: Dijkstra's algorithm, Bellman ford)	**10**

IX	**Hashing** 9.1 Terminologies 9.2 Properties of good hash function 9.3 Hash Functions: Division Function, MID Square, Folding methods. 9.4 Collision resolution techniques: Open addressing: Linear Books: Quadratic probing, Rehashing, Separate chaining 9.5 Case Studies on Non-Linear data structures	**04**

Learning Resources:

1. Brian W. Kernighan and Dennis M. Ritchie: The C Programming Language, Second Edition, Prentice Hall Publication
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
6. Byron S Gottfried, Schaum's Outlines Programming With C, Second Edition, Tata McGraw Hill Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekara, Fundamentals of Computer Algorithms, Galgotia Pub. 2001 ed.

S.Y.B.Sc Computer Science (Semester III)		
CSC-200	**Computer Science Practical – 3** **(Practical on Data Structure and algorithms)**	**Credits: 02** **Hours : 60**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		**Bloom's Cognitive Level**
CO1	Identify and Define various Data Structures (arrays , Stacks, Linked List, Trees , Graphs etc.) needed to solve the problems	**1**
CO2	Demonstrate the searching and sorting algorithms and articulate its application in terms of performance measures	**2**
CO3	Classify the problems into various categories and implement the algorithmic strategies (like Divide and conquer, greedy method, backtracking etc.)	**3**
CO4	Differentiate between types of Data structures studied and analyze which will suit for solving real world problems	**4**
CO5	Compare techniques used for Hashing and Collision Detection	**5**
CO6	Design conventional algorithms and specify techniques used for Traversing Trees and Graphs	**6**

Sr. No.	Title of Experiment / Practical
1.	Structures and Union using array, pointer and functions
2.	Assignments on File handling
3.	Searching Algorithms
4.	Sorting Algorithms
5.	Linked List
6.	Stack
7.	Queue
8.	Assignment on Trees, Tree Traversal Techniques
9.	Application on Trees: Heap Sort
10.	Assignments on Graphs
11.	Assignment on Graph Traversal Techniques
12.	Hashing Methods

S. Y. B. Sc. (Computer Science) - Semester III		
CSC-220	**Basics of Data Analytics (OE-5)**	**Credits: 02 Hours : 30**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's Cognitive Level**
CO1	Identify the type and level of data.	1
CO2	Understand the career opportunities, and future of data analytics	2
CO3	Apply the tools for analysing data and inferring insights	3
CO4	Analyse and handle different types of data and understand data sampling, observation, dataset, prediction, and levels of measurement.	4
CO5	Selection of the specific domain for data analysis	5
CO6	Develop skill set required in data analytics field.	6

Unit No	Contents	No. of Hours
I	**Introduction to Data Analytics** 1.1 What is data analytics? 1.2 Modern Data Ecosystem 1.3 Key Players in the Data Ecosystem 1.4 Need of Data Analytics 1.5 Applications	**03**
II	**Types of Data Analytics** 2.1 Types of data analytics: descriptive, diagnostic, predictive, and prescriptive analytics 2.2 Difference between data analytics and data science	**03**
III	**Dealing with different types of data** 3.1 Terminologies in Data Analytics: Observation, Data Sampling, Dataset and prediction 3.2 Types of Data: Structured, Unstructured and semi-structured 3.3 Qualitative and Quantitative Data 3.4 Data Levels of measurement: Nominal, ordinal, Interval and ratio 3.5 Data Warehousing	**04**
IV	**Tools and techniques for Data Analytics** 4.1 Steps in Data Analysis 4.2 Working with methods for analyzing variety of data 4.3 Working with large data 4.4 Data Visualization using advanced graphs	**10**

V	**Real Life Case Study Discussion** 5.1 Approaches in different domains : Healthcare, Finance, Marketing, Environmental Science	**05**
VI	**Career opportunities, future and case studies** 6.1 Job titles and skills in data analytics (e.g. data scientist, analyst) 6.2 Technology driving future data analytics (e.g. AI, automation) 6.3 Emerging trends in data analytics 6.4 Case Studies	**05**

Learning Resources:

1. Anil Maheshwari, “Data Analytics Made Accessible”, 2020 Edition

Web Resources:

1. www.analyticsvidya.com
2. www.udemy.com
3. https://towardsdatascience.com/machine-learning/home
4. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-advanced-microsoft-excel-for-data-analysis/

S.Y.B.Sc Computer Science (Semester III)		
CSC-230	**Web Page Designing (VSC - Theory)**	**Credits : 02 Hours : 30**
	Course Outcomes (COs) On completion of the course, the students will be able to:	**Bloom's Cognitive Level**
CO1	Describe different web technologies and application development issues and trends	1
CO2	Illustrate the skill and knowledge of Web page design	2
CO3	Apply HTML basics and CSS styles to design the web pages	3
CO4	Explain different elements and its attributes using HTML	4
CO5	Determine the appropriate use of HTML tags and CSS properties to design a web page layout	5
CO6	Design the web site for real-world problems	6

Unit No.	Title of Unit and Contents	No. of hours
I	**Introduction to Internet:** 1.1. World Wide Web 1.2. Web clients, Web server 1.3. Front-end, Back-end terminologies 1.4. Basic Internet protocols 1.4.1. HTTP, HTTPs 1.5. Client Server Architecture 1.5.1. Two-Tier 1.5.2. Multi-Tier 1.5.3. HTTP Request and Response 1.5.4. Understanding the terms: Domain, URL, Hyperlink, web hosting, Browser 1.6 Basic structure of a website	**04**
II	**Introduction to HTML5** 2.1 What is HTML5? 2.1.1 HTML Documents 2.1.2 Basic structure of an HTML document 2.1.3 Creating an HTML document 2.1.4 Mark up Tags 2.2 Heading-Paragraphs 2.2.1 Line Breaks 2.2.2 HTML Tags. 2.3 Elements of HTML 2.3.1 Introduction to elements of HTML 2.3.2 Working with Text 2.3.3 Working with Lists 2.3.4 Tables and Frames	**12**

	2.3.5 Working with Hyperlinks 2.3.6 Images and Multimedia 2.3.7 Working with Forms and controls	
III	**Introduction to Cascading Style Sheets** 3.1 Syntax and selectors 3.2 Ways to insert CSS 3.3 CSS Properties – 3.3.1 Colors, Background, Border, 3.3.2 Margin, Padding, Height and Width, 3.3.3 Text Formatting 3.3.4 List, Tables 3.4 Layout 3.4.1 The display property 3.4.2 The position property 3.4.3 Table layouts	**14**

Learning Resources:

1. DT Editorial Services, "HTML 5 Black Book", Dreamtech Press, 2010
2. Kogent Learning Solutions Inc.,"Web Technologies, Black Book", Dreamtech Press, 2009

e- Resources:

1. https://www.w3schools.com
2. https://geeksforgeeks.com

Skill Sets:

S.Y B. Sc. (Computer Science) – Semester III		
CSC-240	**Software Engineering Approach Using UML (SEC)**	**Credits: 02 Hours: 30**
Students will acquire the following skills on completion of the course:		
1.	Understanding of Software Concepts: Ability to define and explain fundamental concepts related to software engineering, including software, software processes, and software development methodologies.	
2.	Understanding the fundamental principles of object orientation and proficiency in defining and using objects, classes, instances, and specifying attributes with visibility.	
3.	Capability to perform object-oriented analysis to identify system requirements.	
4.	Skill in identifying and defining classes and objects.	
5.	Understanding the concepts of UML (Unified Modeling Language) and Proficiency in creating and interpreting use case diagrams.	
6.	Skill in creating and interpreting class diagrams and ability to define relationships and common mechanisms in a class.	
7.	Capability to work with advanced classes and relationships and knowledge of interfaces, types, roles, and packages.	
8.	Proficiency in creating and interpreting interaction diagrams, sequence diagrams, activity diagrams, collaboration diagrams, and state chart diagrams.	
9.	Skill in creating and interpreting component diagrams, components, and deployment diagrams.	
10.	Understanding software development requirements, collaborating with team members, and presenting project ideas and results.	

S.Y B. Sc. (Computer Science) – Semester III		
CSC-240	**Software Engineering Approach Using UML (SEC- Theory)**	**Credits : 02 Hours : 30**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's Cognitive Level**
CO1	Define the fundamental concepts of software engineering, including software, software processes, and software development methodologies	1
CO2	Discuss UML for System Modeling	2
CO3	Demonstrate the use of class based model for real-world case studies	3
CO4	Analyze Behavioral Modeling Strategies	4
CO5	Evaluate architectural models based on requirements	5
CO6	Prepare UML for utility-based projects	6

Unit No.	Title of Unit and Contents	No. of hours
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 Process Models-A Generic Process Model, 1.6.1 Prescriptive Process Models- Waterfall Model, Incremental Process Model 1.6.2 Evolutionary Process Models- Prototyping Model, Spiral Model, Concurrent Model	**06**
II	**Requirements Analysis and UML** 2.1 Requirement Engineering Tasks – Inception, Elicitation, Elaboration, Negotiation, Specification, Validation 2.2 Software Requirement Specification (SRS) 2.3 Structured Analysis with DFD 2.4 Unified Modelling Language 2.2.1 Advantages and Features of UML 2.2.2 UML Diagrams (Two case studies should be covered)	**08**

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 (Minimum Two case studies should be covered on each diagram)	**10**
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 Behavioral 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 (Minimum Two case studies should be covered on each diagram)	**06**

Learning Resources:

1. Roger S. Pressman, Software Engineering : A Practitioner's Approach, McGraw Hill International Edition, 7th Edition
2. Grady Booch, James Rambaugh, The Unified Modeling Language User / Reference Guide, Pearson Education INC
3. Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC Craig Larman, Applying UML and Patterns, Pearson Education INC

Proposed Evaluation Methods:

1. Case Study
2. Problem Solving
3. Report Writing

4. Presentation / Poster presentation
5. Seminars

S.Y.B.Sc Computer Science (Semester III)		
CEP-245	**Foundations of Community Engagement (CEP)**	**Credits: 02 Hours : 30**

Community engagement –Basics (1 Credit)	
Topics Covered	**Activities**
Introduction to Community Engagement	- Overview of theories and models - Importance of interdisciplinary approaches
Social Issues Analysis	- Guest lecture by a social scientist or experts from diverse sectors - Group discussion and analysis of contemporary social issues
Community Needs Assessment	- Theory on needs assessment methodologies - Field visit for practical application
Stakeholder Engagement	- Guest lecture from a community organizer - Simulated stakeholder engagement role-play
Community engagement –Field Work (1 Credit)	
Topics Covered	**Activities**
Cultural Competence in Community Work	- Cultural sensitivity training - Case studies on community engagement
Writing Project Proposal and finance resource management	- Develop a community project proposal and finance resource management - Timeline for implementation
Field Work Skills Training	- Training in data collection, interviewing, and observation - Practical exercises in the community
Ethical Considerations in Community Engagement	- Guest lecture on ethical dilemmas in community work - Case studies and group discussions

Community Engagement and Social Responsibility (CESR) Course

The Community Engagement and Social Responsibility course is an immersive and transformative learning experience designed for second-year undergraduate students. In an era where the intersections of diverse disciplines are more critical than ever, this course stands at the forefront of transdisciplinary and multidisciplinary education. As the heartbeat of societal progress, this compulsory course seeks to connect students with their communities, fostering a deep sense of social responsibility. Rooted in the belief that academic knowledge should transcend classroom walls, the aim is to equip students with the tools to analyse, comprehend, and address pressing social issues. Through dynamic and interactive learning methods, students will not only explore the complexities of community dynamics but also actively contribute to the development of sustainable solutions.

Objectives:

1. CESR Theory
   a) To develop an understanding of community needs and challenges.

b) To equip students with skills to identify problem areas within the community.

c) To guide students in creating effective project proposals.

d) To apply classroom knowledge of courses to field realities and thereby improve the quality of learning.

2. CESR Field Work:

a) To provide practical experience in implementing community projects.

b) To assess students' ability to apply theoretical knowledge in real-world situations.

c) To develop skills in project management, teamwork, and communication.

S.Y.B.Sc Computer Science (Semester III)		
MTS--215	**Applied Linear Algebra (Minor -Theory)**	**Credits: 02 Hours : 30**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's Cognitive Level**
CO1	Recall all Matrix operations and properties	1
CO2	Discuss concepts of linear independence, spanning set, basis, orthogonality	2
CO3	Compute inner product, norm, angle, distance between vectors, Eigenvalues, Eigenvectors of matrices.	3
CO4	Explain Linear transformations and its basic properties	4
CO5	Determine whether the matrix is diagonalizable.	5
CO6	Develop ability to apply linear algebra concepts to solve problems in various fields.	6

Unit No.	Title of Unit and Contents	No. of hours
I	**Matrices and Linear Equations** 1.1 Introduction 1.2 Matrices 1.3 Elementary Row operations, Row Echelon form. 1.4 Solution to System of Linear Equations	4
II	**Vector Spaces** 2.1 Introduction 2.2 Euclidean Spaces 2.3 Subspaces 2.4 Linear Span 2.5 Linear Independence 2.6 Basis 2.7 Coordinates	9

III	**Inner Product** 3.1 Introduction 3.2 Length, Distance, Angle 3.3 Orthogonality 3.4 Gram-Scmidt Orthogonalization Process	4
IV	**Linear Transformations** 4.1 Introduction 4.2 Linear transformation 4.3 Kernel and Range of a Linear Transformation 4.4 Standard Matrix	7
V	**Eigenvalues and Eigenvectors** 5.1 Introduction 5.2 Eigenvalues and Eigenvectors 5.3 Diagonalisation	6

Learning Resources:

1) Elementary Linear Algebra with supplemental Applications, by Howard Anton, Chirs Rorres, Wiley Student Edition, Eleventh Edition.

2) Linear Algebra and it's Applications, David C. Lay, Steven R. Lay, Judi J. MacDonald Pearson Publication, 2016, Fifth Edition.

3) Linear Algebra with Applications, W. Keith Nicholson, Lyryx Learning Team.

S.Y.B.Sc Computer Science (Semester III)		
MTS-216	**Mathematics Practical- 3**	**Credits: 02** **Hours : 30**
Course Outcomes (COs) **Completion of the course, the students will be able to:**		**Bloom's Cognitive level**
CO1	Show different matrix operations in python.	1
CO2	Explain different Numerical Interpolation techniques.	2
CO3	Illustrate Basic programming skills in python.	3
CO4	Explain different concepts related to vectors.	4
CO5	Evaluate EIgenvalues and Eigenvectors of matrices.	5
CO6	Development of problem-solving skills by applying mathematical concepts to practical situations.	6

Unit No.	Title of Unit and Contents
1	Introduction to python - I (Data Types, List, tuples, array)
2	Introduction to python - II (basic operations on matrices)
3	Basic Python Programming - I
4	Basic Python Programming - II
5	Newton's Forward Interpolation Technique using Python
6	Newton's Backward Interpolation Technique using Python
7	Divided Difference Interpolation Technique using Python
8	Lagrange Interpolation Technique using Python
9	Vector Spaces
10	Inner product spaces
11	Eigenvalues and Eigen Vectors - I
12	Eigenvalues and Eigen Vectors - II
13	Student Activity –I
14	Students Activity – II
15	Students Activity – III

S.Y B.Sc. (Computer Science) – Semester III		
ELS - 215	**Embedded Systems (Minor-Theory)**	**Credits: 02 Hours : 30**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Blooms cognitive level**
CO1	Describe the hardware and software unit of Embedded System.	1
CO2	Explain the design process of Embedded system.	2
CO3	Demonstrate the interfacing peripheral devices to Embedded Controller	3
CO4	Categorize different sensors as per the need of the application.	4
CO5	Explain the interfacing techniques for embedded system.	4

Unit No.	Title and contents	No. of hours
I	**Hardware and software units of Embedded system** 1.1 Definition of embedded system 1.2 Classification of embedded systems 1.3 Processors of embedded into a system 1.4 Hardware Units of embedded system 1.5 Software embedded into a system 1.6 Architecture of Embedded System 1.7 Design process in embedded system	(08)
II	**Architecture of Embedded Controller** 2.1 Introduction to RISC and CISC 2.2 Microcontroller architecture 2.3 Counter/ timers and interrupts 2.4 Serial communication 2.5 Addressing modes and instruction set 2.6 Assembly language programs 2.7 I/O Interfacing	(10)

III	**Sensors, Data converters, Actuators and Interfacing** 3.1 Classification of sensors (Active and passive, analog and digital), 3.2 Specifications of sensor: Accuracy, range, linearity, sensitivity, resolution, reproducibility, 3.3 Temperature sensors (LM-35, AD590), piezoelectric humidity sensor (DHT11), optical sensor (LDR), displacement sensor (LVDT), Passive Infrared sensor (PIR), Touch sensor, Ultrasonic sensor, Gas Sensor 3.4 Analog to Digital and Digital to Analog Converter 3.5 Actuators: LED, solenoid, Motor, servomotor, LCD, OLED, TFT, electronic switches and relay 3.6 Interfacing Techniques: Parallel, serial	(12)

Learning Resources:

1. Embedded Systems SoC,IoT, AI and Real – Time Systems, 4th edition: Raj Kamal, MacGrawHill Education (India) Pvt. Ltd.
2. Embedded Systems Architecture, Programming and Design, 2nd edition: Raj Kamal, MacGrawHill Education (India) Pvt. Ltd.
3. Sensors & Transducers: Dr. A. D. Shaligram: CTC publications
4. The 8051 microcontroller and embedded systems using assembly and C, second edition, Muhammad Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay, Pearsons Publication

S.Y B.Sc. (Computer Science) – Semester III		
ELS – 216	**Electronics Practical -3 (Minor- Practical)**	**Credits: 02 Hours : 60**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Identify components and devices used in the circuit.	2
CO2	Use instruction set of microcontroller to write and execute assembly language programs.	3
CO3	Use embedded C /equivalent language for interfacing different devices to microcontroller.	3
CO4	Implement the understanding of architecture of microcontroller to interface different Input / Output devices.	3
CO5	Test embedded C /equivalent language programs for interfacing I/O devices to microcontroller.	4
CO6	Relate the output generated from the interfacing I/O devices to microcontroller.	4

Sr. No.	Title of Experiment / Practical
1	Introduction to Integrated Development Environment
2	Arithmetic problems using Assembly language programming
3	Logical problems using Assembly language programming
4	Testing Boolean operations using Assembly language programming
5	Testing data transfer operations using Assembly language programming
6	Testing program branching operations using Assembly language programming
7	Code conversion problems using Assembly language programming
8	Blinking of LEDs interfaced to microcontroller.
9	Traffic light control by microcontroller.
10	Interfacing Seven Segment Display with microcontroller.
11	Interfacing LCD with microcontroller.
12	Speed and Direction Control of stepper motor interfaced to microcontroller.
13	Interfacing a switch with microcontroller to control Bulb using relay
14	Square Wave generation Using microcontroller
15	Triangular Wave generation Using microcontroller
16	Or Any Other Equivalent Experiment

S.Y B.Sc. (Computer Science) – Semester III		
STS -215	**Statistical Methods I** **(Minor-Theory)**	**Credits: 02** **Hours : 30**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		**Bloom's cognitive level**
CO1	Define various discrete probability distributions and outline the properties of probability mass functions, cumulative distribution functions.	1
CO2	Distinguish between multiple and partial correlation.	2
CO3	Demonstrate multiple regression model.	3
CO4	Relate the bivariate probability distributions to real life situations.	4
CO5	Measure partial regression coefficient, multiple and partial correlation coefficient for tri-variate data.	5
CO6	Write an equation of plane of regression for the given data.	6

Unit No.	Title of Unit and Contents	No. of hours
I	**Multiple and Partial Correlation and Regression (for trivariate data)** 1.1 Yule's notation and concept of multiple regression 1.2 Fitting of multiple regression plane 1.3 Partial regression coefficient, interpretation 1.4 Multiple correlation coefficient, concept, definition, computation and interpretation 1.5 Partial correlation coefficient, concept, definition, computation and interpretation 1.6 Numerical Problems	06
II	**Discrete Random variable** 2.1 Definition of random variable and discrete random variable 2.2 Definition of probability mass function 2.3 Distribution function and its properties 2.4 Definition of expectation and variance 2.5 Theorem on expectation 2.6 Determination of median and mode using p.m.f, 2.7 Numerical problems related to real life situations.	04
III	**Standard Discrete Probability Distributions** 3.1 Discrete Uniform Distribution: definition, mean, variance 3.2 Bernoulli Distribution: definition, mean, variance,	14

	additive property 3.3 Binomial Distribution: definition, mean, variance, additive property 3.4 Geometric Distribution (p.m.f p(x) = pqX, x = 0,1,2.....): definition, mean, variance 3.5 Poisson Distribution: definition, mean, variance, mode, additive property, limiting case of B (n, p) 3.6 Illustration of real life situations 3.7 Numerical problems related to real life situations.	
IV	**Bivariate discrete probability distribution** 4.1 Definition of two-dimensional discrete random variable, its joint p.m.f. and its distribution function and their properties 4.2 Concept of identically distributed random variables 4.3 Computation of probabilities of events in bivariate probability distribution 4.4 Concepts of marginal and conditional probability distributions 4.5 Independence of two discrete random variables based on joint and marginal p.m.f, 4.6 Examples and problems.	06

Learning Resources:

1. Statistical Methods, G.W. Snedecor, W.G. Cochran, John Wiley & sons, 1989.
2. Fundamentals of Applied Statistics (4th Edition), Gupta and Kapoor, S. Chand and Sons, New Delhi, 2014.
3. Modern Elementary Statistics, Freund J.E., Pearson Publication, 2005.
4. A First course in Probability 6th Edition, Ross, Pearson Publication, 2006.

S.Y B.Sc. (Computer Science) – Semester III		
STS -216	**Statistics Practical - 3 (Minor- Practical)**	**Credits: 02 Hours : 60**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Recall the concepts of tri-variate data, multiple and partial correlation coefficient, partial regression coefficient and its interpretation.	1
CO2	Discuss various applications of statistical measures using R software.	2
CO3	Execute the computational techniques using R software.	3
CO4	Analyse different concepts of statistics using R software.	4
CO5	Validate the fundamental knowledge and represent using R software.	5
CO6	Write a program using R to build plane of regression for the given data.	6

Sr. No.	Title of Experiment / Practical
1	Multiple Regression I
2	Multiple Regression II
3	Discrete Probability theory
4	Applications of Binomial distribution
5	Fitting of Binomial distribution
6	Applications of Poisson distribution
7	Fitting of Poisson distribution
8	Applications of Geometric distribution
9	Bivariate Probability theory
10	Multiple Regression using R
11	Computations of probabilities using R-I (Binomial)

12	Computations of probabilities using R-II(Poisson)
13,14&15	Applications of Statistical techniques to real-life data.

Fergusson College (Autonomous), Pune

Learning Outcomes-Based Curriculum

for 3/4 years B**.** Sc. /B. Sc. (Honours) Programme

as per guidelines of

NEP-2020

for

S.Y B. Sc. (Computer Science)

SEMESTER IV

With effect from Academic Year

2024-2025

S.Y.B.Sc Computer Science (Semester IV)		
CSC-251	**Object Oriented Programming using C++**	**Credits: 04 Hours : 60**
	Course Outcomes (COs) On completion of the course, the students will be able to:	**Bloom's Cognitive Level**
CO1	Understand object-oriented concepts.	1
CO2	Illustrate and use the basic programming constructs of C++	2
CO3	Apply C++ data type such as arrays, strings, and pointers	3
CO4	Analyse memory utilization, including proper allocation/deallocation methods.	4
CO5	Test and perform critical evaluation of the program outcome to validate the Correctness of the algorithm.	5
CO6	Integrate various object-oriented approaches to solve software problems in C++	6

Unit No.	Contents	No. of Hours
I	**Object oriented concepts** 1.1 Object oriented concepts : Class, object, Abstraction, Encapsulation, Inheritance, Polymorphism 1.2 Difference between C and C++ 1.3 Features, advantages and Applications of OOPs	03
II	**Introduction to C++** 2.1 Data types, new operators and keywords, using namespace concept 2.2 Simple C++ Program 2.3 Introduction to Reference variables 2.4 Usage of 'this' pointer 2.5 Classes and Objects 2.6 Access specifiers 2.7 Defining Data members and Member functions 2.8 Array of objects	10
III	**Function in C++ , Constructors and destructor** 3.1 Call by reference, Return by reference 3.2 Function overloading and default arguments 3.3 Inline function 3.4 Static class members 3.5 Friend Concept – Function, Class	10

	3.6 Types of constructors 3.7 Memory allocation (new and delete) 3.8 Destructor	
IV	**Operator overloading** 4.1 Overloading Unary and Binary operators, manipulation of strings using operators 4.2 Overloading using friend function 4.3 Type casting and Type conversion	06
V	**Inheritance** 5.1 Types of inheritance with examples 5.2 Constructors and destructor in derived classes 5.3 Virtual base classes, Virtual functions and Pure virtual function 5.4 Abstract base classes	08
VI	**Managing Input and Output using C++** 6.1 Managing console I/O 6.2 C++ stream classes 6.3 Formatted and unformatted console I/O 6.4 Usage of manipulators	03
VII	**Working with files** 7.1 File operations – Text files, Binary files 7.2 File stream class and methods 7.3 File updation with random access 7.4 Overloading insertion and extraction operator	07
VIII	**Templates** 8.1 Introduction to templates 8.2 Class templates, function templates and overloading of function templates 8.3 Templates with multiple parameters	07
IX	**Exception Handling in C++** 9.1 try, catch and throw primitives	04
X	**Applications of C++**	02

Learning Resources:

1. Object Oriented Programming in C++ by Robert Lafore Techmedia Publication
2. The C++ programming language by Bjarne Stroustrup
3. Object oriented programming with C++. By E.Balagurusamy
4. Object Oriented Programming in C++ R Rajaram New Age International Publishers 2nd

S.Y.B.Sc Computer Science (Semester IV)		
CSC-250	**Computer Science Practical -4 (Major-Practical)**	**Credits : 02 Hours : 60**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Define objects, classes and understand the given problems in C++.	1
CO2	Illustrate the use of constructors, data types, operators, control structures and functions in C++ programming.	2
CO3	Implement and execute the programs based on overloading concepts	3
CO4	Compare and contrast the types of inheritance to implement real-world problems	4
CO5	Test and validate the programs	5
CO6	Design the programs to implement the concept such as functions, inheritance, template and exception in C++	6

Sr. No.	Title of Experiment / Practical
1.	Classes and Objects
2.	Constructor and Destructor
3.	Inline function, friend function, default argument
4.	Function Overloading
5.	Operator overloading
6.	Inheritance
7.	Formatted Input/ Output
8.	File Handling
9.	Template
10.	Exception handling
11.	Activity-1
12.	Activity-2

S.Y B.Sc Computer Science (Semester IV)		
CSC-270	**e-Commerce (OE-6)**	**Credits : 02 Hours : 30**
	Course Outcomes (COs) On completion of the course, the students will be able to:	**Bloom's Cognitive Level**
CO1	Describe internet trading relationships	1
CO2	Explain legal and privacy issues in e-commerce	2
CO3	Demonstrate the use of business models for real time case studies	3
CO4	Analyse the impact of e-commerce on business models and strategy	4
CO5	Compare the performance of electronic payment systems	5
CO6	Create in-bound and out-bound logistics for supply chain management	6

Unit No.	Title of Unit and Contents	No. of hours
I	**Introduction to e-Commerce** 1.1 The Scope of Electronic Commerce, Definition of Electronic Commerce 1.2 Electronic Commerce and the Trade Cycle 1.3 Electronic Markets, 1.4 Electronic Data Interchange, 1.5 Internet Commerce, e-Commerce in Perspective	06
II	**Business Strategy in an Electronic Age** 2.1 Business Strategy in an Electronic Age 2.2 Supply Chains, Porter's Value Chain Model 2.3 Inter-Organizational Value Chains, Competitive Strategy, 2.4 Porter's Model, First Mover Advantage, Sustainable, 2.5 Competitive Advantage, 2.6 Competitive Advantage using e-Commerce, Business Strategy	08
III	**Business-to-Business Electronic Commerce** 3.1 Business-to-Business Electronic Commerce 3.2 Characteristics of B2B EC, Models of B2B EC 3.3 Procurement Management Using the Buyer's Internal Marketplace, Supplier-Oriented Marketplace, Intermediary-Oriented Marketplace 3.4 Just-in-Time Delivery 3.5 Auctions and Services from Traditional to Internet-Based EDI 3.6 Integration with Back-end Information Systems 3.7 The Role of Software Agents for B2B EC 3.8 Electronic Marketing in B2B 3.9 Solutions of B2B EC, Managerial Issues	08

IV	**Electronic Payment Systems** 4.1 Schemes in Electronic Payment Systems, Electronic Credit 4.2 Card System on the Internet, Electronic Fund Transfer and 4.3 Debit Cards on the Internet, Stored-Valued Cards and E-Cash 4.4 Electronic Check Systems, Prospect of Electronic Payment 4.5 Systems, Public Policy, From Legal Issues to Privacy 4.6 EC-Related Legal Incidents, Legal, Ethical, Protecting 4.7 Privacy, Protecting Intellectual Property	08

Learning Resources:

1. Bharat Bhasker, "Electronic Commerce Framework, Technologies and Applications", McGraw Hill, 4th Edition, 2014
2. David Whiteley, "e-Commerce", Tata McGraw Hill, 2000.
3. Eframi Turban, Jae Lee, David King, K. Michale Chung, "Electronic Commerce", Pearson Education, 2000

e- Resources:

1. https://www.w3schools.com
2. https://geeksforgeeks.com

S.Y.B.Sc Computer Science (Semester IV)		
CSC-280	**Advanced Web Page Designing (VSC-Theory)**	**Credits : 02 Hours : 30**
	Course Outcomes (COs) On completion of the course, the students will be able to:	**Bloom's Cognitive Level**
CO1	Describe different JavaScript popup boxes , fundamentals of JavaScript and XML	1
CO2	Explain looping and functions in JavaScript.	2
CO3	Illustrate the use of objects in JavaScript.	3
CO4	Compare HTML and XML.	4
CO5	Determine various event handling methods.	5
CO6	Design the web site for real-world problems.	6

Unit No.	Title of Unit and Contents	No. of hours
I	**Introduction to JavaScript :** 1.1 What is JavaScript? 1.2 Exploring Popup boxes 1.2.1 Alert 1.2.2 Confirm 1.2.3 Prompt 1.3 Linking JavaScript 1.1.1 In Head element 1.1.2 In Body element 1.1.3 Using external file	**03**
II	**Exploring fundamentals of JavaScript** 2.1Decision making statements 2.1.1 Variable 2.1.2 Operators 2.1.3 If and nested if conditions 2.2Looping statements 2.2.1 while loop 2.2.2 do while loop 2.2.3 for loop 2.3 Functions in JavaScript 2.3.1 Defining and calling a function 2.3.2 Defining function arguments 2.3.3 Defining return statement	**08**
III	**Objects and event handling in JavaScript** 3.1 What is object? 3.1.1 Array object 3.1.2 Math object 3.1.3 Date object 3.1.4 Document object	**14**

	3.2 Event handling 3.2.1 Managing JavaScript events 3.2.2 Mouse events 3.2.3 Keyboard events	
IV	**Introduction to XML** 4.1 XML introduction 4.2 XML features 4.3 XML versus HTML 4.4 XML examples 4.5 XML attributes 4.6 XML validation 4.6.1 Valid XML document 4.6.2 Rules for well-formed XML	**05**

Learning Resources:

1. DT Editorial Services, "HTML 5 Black Book", Dreamtech Press, 2010
2. Kogent Learning Solutions Inc.,"Web Technologies, Black Book", Dreamtech Press, 2009
3. O'Reilly, "Learning XML", Eric T. Ray, 2001

e- Resources:

1. https://www.w3schools.com
2. https://geeksforgeeks.com

Skill Sets:

S.Y.B.Sc Computer Science (Semester IV)		
CSC-290	**Fundamental of Software Testing (SEC)**	**Credits: 2 Hours: 30**
Students will acquire the following skills on completion of the course:		
1.	Understand Software Testing, its principles and Software Testing Life Cycle	
2.	Learn the Defect Management Process and Define the Defect	
3.	Get Familiar with testing methodologies such as black box testing, white box testing, and gray box testing	
4.	Gaining knowledge of different types of testing including functional testing, non-functional testing (performance, usability), and regression testing	
5.	Apply thinking and analytical skills for writing test cases, pre-conditions and post-conditions for different case studies	
6.	Ability to write and execute test cases, test scripts, and test scenarios and Validate the test cases on various case studies	
7.	Ability to analyze test results and provide meaningful reports to stakeholders	
8.	Proficiency in test case design techniques such as boundary value analysis, equivalence partitioning, and decision tables	
9.	Apply different Automation Tools to write test cases of real-world problems	
10.	Learn to detect the issues in any software applications and Generate Report on the same	

S.Y.B.Sc Computer Science (Semester IV)		
CSC-290	**Fundamentals of Software Testing (SEC- Theory)**	**Credits: 02 Hours : 30**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Describe the need of software testing and list different types of defects.	1
CO2	Discuss the basic concepts of testing techniques and illustrate various types of testing	2
CO3	Apply software testing skills in different domains and develop test plans for evaluation	3
CO4	Detect the issues in software applications	4
CO5	Select appropriate automation tool for testing and analyze its effectiveness	5
CO6	Write the test cases to improvise the efficiency of the application	6

Unit No.	Contents	No. of Hours
I	**Software Testing Fundamentals** 1.1 Software testing- Definition, 1.2 Importance of Testing, 1.3 Role of Tester, 1.4 Software Testing Principles, 1.5 Software Testing Life Cycle 1.6 Defects - Definition and types of defects, 1.7 Defect Management Process, 1.8 Defect/Bug Life Cycle	**06**
II	**Types of Testing** 2.1 Manual Testing- Black Box testing, White Box testing, 2.2 Unit testing, 2.3 System testing, 2.4 Integration testing 2.5 Acceptance testing 2.6 Automation testing- Automation Testing 2.7 Automated Testing Process 2.8 Test tool selection 2.9 Framework for Automation 2.10 Types of Automated Testing 2.11 Regression Testing 2.12 Non-Functional Testing 2.13 Automation Testing Tools – Different tools	**08**
III	**Test Case Development**	**06**

	3.1 Overview of Test Documentation 3.2 Writing Test Cases 3.3 Test Analysis 3.4 Requirements Traceability Matrix (RTM) 3.5 Test Data Generation: What, How, Example, Tools	
IV	**Testing Techniques** 4.1 Software Testing Techniques with Test Case Design 4.2 Boundary Value Analysis 4.3 Equivalence Partitioning 4.4 Decision Table Testing 4.5 State Transition Testing 4.6 Use Case Testing 4.7 Agile Methodology and Scrum Testing Methodology	**05**
V	**Testing Different Domains** 5.1 Web Application Testing, 5.2 Finance domain Application testing, 5.3 e-Commerce testing 5.4 Healthcare 5.5 one case study of each domain	**05**

Learning Resources:

1. Srinivasan Desikan, "Software Testing Principals and practices", Pearson Publication ISBN-13 978-8-17-758295-6, 2013
2. Glenford J. Myers, Corey Sandler, Tom Badgett, "The Art of Software Testing", 3rd Edition ISBN: 978-1-118-13315-6, 2011
3. Kshirasagar Naik And Priyadarshi Tripathy, "Software testing and quality assurance: Theory and Practice", A John Wiley & Sons, Inc., Publication, ISBN 978-0-471-78911-6, 2008

Web Contents-

1. www.tutorialspoint.com
2. https://www.javatpoint.com/software-testing-tutorial
3. https://www.guru99.com/software-testing.html

e-resources-

1. http://epathshala.nic.in/
2. https://www.coursera.org/
3. https://inflibnet.ac.in/

Proposed Evaluation Methods:

1. Case Study to Apply testing technique to solve real-world problems
2. Report writing For the same
3. Presentation / Poster presentation

S.Y.B.Sc Computer Science (Semester IV)		
FP-295	**Community Engagement - Field Project (FP)**	**Credits: 02 Hours : 30**
Foundations of Field Work (1 credit)		
Topics Covered	**Activities**	
Field visits, Field work Reflection and Analysis	- Reflective journals on field experiences - Group presentations	
Community Impact Assessment	- Methods for assessing project impact - Group project: Conduct impact assessment in a chosen community	
Advanced Field Work (1 credit)		
Topics Covered	**Activities**	
Field Work, Project Presentation Review and Integration	- Review of key concepts from previous credits - Integration of community engagement and fieldwork principles - Analysis - Submission of CEP/FP project report	

<u>Evaluation consist of two parts:</u>

Evaluate each student for 50 marks per semester at department level –

- 20 marks for Continuous evaluation (CE)
  - Progress report on project implementation. (Field diary)

- 30 marks for End Semester Examination (ESE)
  - Project Report
  - Final presentation of field project findings assessing project outcomes and reflections.

S.Y.B.Sc Computer Science (Semester IV)		
MTS-265	**Computational Geometry (Minor-Theory)**	**Credits:02 Hours : 30**
Course Outcomes (COs)		
	Description on completion of the course, the students will be able to:	**Bloom's Cognitive level**
CO1	Identify basic 2-D and 3-D transformation matrices like Shearing, Scaling reflections, rotations.	1
CO2	Understand the effect of transformations on the points, intersecting lines, parallel lines.	2
CO3	Apply different types of projections.	3
CO4	Explain rotation about arbitrary point in 2D and about arbitrary axis in 3D	4
CO5	Determine reflections through arbitrary lines in 2D and arbitrary planes in 3D.	5
CO6	Develop ability to design and analyze algorithms for solving geometric tasks.	6

Unit No.	Title of Unit and Contents	No. of hours
I	**Two Dimensional Transformations** 1.1 Basic 2-D transformations (Scaling, Shearing, Rotation about origin, Reflections) 1.2 Transformation of points, Straight lines 1.3 Solid body transformations 1.4 Concatenation of transformations 1.5 Rotation about arbitrary point 1.6 Reflection through an arbitrary line	12
II	**Three Dimensional Transformations** 2.1 Basic 3-D transformations 2.2 Concatenation 2.3 Rotation about an axis parallel to any one of the coordinate axes 2.4 Reflection through a plane parallel to anyone of the coordinate planes 2.5 Rotation about an arbitrary axis 2.6 Reflection through an arbitrary plane	10
III	**Projection** 3.1 Introduction 3.2 Orthographic Projections 3.3 Axonometric Projections 3.4 Oblique Projections 3.5 Single point Perspective Projections	08

Learning Resources:

1) D.F.Rogers, J.A.Adams,Mathematical elements for Computer Graphics,McGraw Hill Edition.
2) Duncan Marsh, Applied Geometry for Computer Graphics and CAD, Springer Publication, Second Edition.
3) M.E.Mortenson,Computer Graphics Handbook, Industrial Pres Inc.

S.Y.B.Sc Computer Science (Semester IV)		
MTS-266	**Mathematics Practical- 4 (Minor Practical)**	**Credits : 02 Hours : 30**
Course Outcomes (COs)		
	Description on completion of the course, the students will be able to:	**Bloom's Cognitive level**
CO1	Show different geometric effects using Python.	1
CO2	Understand 2-D and 3-D transformations.	2
CO3	Apply different types of projections.	3
CO4	Classify the position of a point with respect to line/convex polygon.	4
CO5	Evaluate equispaced points on different plane curves.	5
CO6	Develop ability to design and analyze algorithms for solving geometric tasks.	6

Unit No.	Title of Unit and Contents
1	Generate n- equidistant points on a circle.
2	Generate n- equidistant points on a Ellipse.
3	Generate n- equidistant points on a parabola y^2= 4 a x
4	Generate n- equidistant points on a hyperbola.
5	2 Dimensional transformations
6	3 Dimensional transformations
7	Projections
8	Implementation of 2D and 3D transformations using python-I
9	Implementation of 2D and 3D transformations using python-II
10	Sorting of points with respect to line (using python programming)Sorting of points with respect to convex polygon(using python programming)
11	Finding the pairs of points having shortest mutual distance and maximum mutual distance.
12	Find the nearest neighborhood of each point in the given set.
13	Student Activity –I
14	Students Activity – II
15	Students Activity – III

S.Y B.Sc. (Computer Science) – Semester IV		
ELS - 265	**Fundamentals of Internet of Things (IoT) (Minor-Theory)**	**Credits: 02 Hours : 30**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Define IoT.	1
CO2	Discuss the basic IoT reference model.	2
CO3	Use of communication devices for IoT.	3
CO4	Analyze IoT systems.	4
CO5	Compare IoT enabling Technologies.	4
CO6	Design of M2M architecture.	6

Unit No.	Unit Title and Contents	No. of hours
I	**Architecture of Internet of Things(IoT)** 1.1 Definition of IoT 1.2 History of IoT 1.3 Examples of IoT 1.4 Single Board Computer 1.5 Architecture of IoT: Components, six-layer architecture model 1.6 Data Collection and storage 1.7 Cloud Platforms 1.8 IoT communication protocols: wireless, wired communication protocols, comparison of communication technologies 1.9 Internet connectivity	(12)
II	**IoT connectivity Technologies** 2.1 Introduction 2.2 IEEE 802.15.4 2.3 ZigBee 2.4 Bluetooth 2.5 LoRa 2.6 RFID 2.7 NFC 2.8 NB-IoT 2.9 Sigfox 2.10 Wifi	(10)
III	**IoT enabling Technologies** 3.1 Wireless Sensor network 3.2 M2M Architecture, 3.3 Basic Nodal Capabilities 3.4 Industrial Internet of Things (IIoT) 3.5 Cyber Physical System (CPS)	(08)

Learning Resources:

1. Simone Cirani, Gianluigi Ferrari, Marco Picone, Luca Veltri. Internet of Things: Architectures, Protocols and Standards, 1 st edition, Wiley Publications, 2019.
2. Bahga, Arshdeep, and Vijay Madisetti. Internet of Things: A hands-on approach, 1st edition, University press, 2014
3. Vermesan, Ovidiu, and Peter Friess, eds. Internet of things-from research and innovation to market deployment, 1st edition, Aalborg: River publishers, 2014.
4. Tsiatsis, Vlasios, Tsiatsis, Vlasios, Stamatis Karnouskos, Jan Holler, David Boyle, and Catherine Mulligan, Internet of Things: technologies and applications for a new age of intelligence, 2nd edition, Academic Press, 2018.

S.Y B.Sc. (Computer Science) – Semester IV		
ELS - 266	**Electronics Practical – 4 (Minor- Practical)**	**Credits: 02 Hours : 60**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Identify components and devices used in the circuit.	2
CO2	Use of actuators.	3
CO3	Demonstrate the working of sensor.	3
CO4	Implement the IoT connectivity technologies	3
CO5	Test programming techniques for data transfer to cloud. .	5
CO6	Design web page to control devices.	6

Sr. No.	**Title of Experiment / Practical**
1	Hands on Python/HTML programming (Data types, Operators)
2	Hands on Python/HTML programming (Function handling)
3	Internet Controlled LEDs
4	DHT 11/22 temperature and humidity monitoring system
5	LDR to control ON/OFF action of LED
6	Distance measurement using ultrasonic sensor HC-SR04
7	Human motion detection using PIR sensor
8	Soil Moisture measurement
9	Clockwise/anticlockwise rotation of stepper motor/servomotor
10	DC motor interfacing for smart irrigation
11	Data transfer using ZigBee protocol
12	Long range data transmission using LoRa
13	Authentication approval/denial using RFID interfacing

14	Web page design for IoT application
15	Sensor Data transfer and retrieval to and from cloud
16	Or Any Other Equivalent Experiment

S.Y B.Sc. (Computer Science) – Semester IV		
STS - 265	**Statistical Methods II (Minor-Theory)**	**Credits: 02 Hours : 30**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Define various continuous probability distributions and outline the properties of probability density functions, cumulative distribution functions.	1
CO2	Explain basic models of time series and different methods of estimation of trend and seasonal variation.	2
CO3	Demonstrate the significance of the distributions and identify the real -life situations for probability distributions.	3
CO4	Relate the probability distributions to real life situations.	4
CO5	Determine an appropriate model to forecast future observations of the time series.	5
CO6	Build an appropriate time series model for the given data.	6

Unit No.	Title of Unit and Contents	No. of hours
I	**Continuous Random Variable** 1.1 Definition of continuous random variable (r.v.) 1.2 Probability density function (p.d.f.) 1.3 Cumulative distribution function (c.d.f.), its properties, 1.4 Calculation of mean, mode, median, variance 1.5 Standard deviation for continuous r. v. 1.6 Numerical problems related to real life situations.	04
II	**Standard Continuous Probability Distributions** 2.1 Uniform Distribution: statement of p.d.f., mean, variance 2.2 Nature of probability curve 2.3 Exponential Distribution: statement of p.d.f. of the form, $f(x) = (1/\theta)\, e(-x/\theta)$, mean, variance, nature of probability curve, lack of memory property 2.4 Normal Distribution: statement of p.d.f., identification of parameters, nature of probability density curve, standard normal distribution, symmetry, distribution of aX+b, aX+bY+c where X and Y are independent normal variables, computations of probabilities using normal probability table 2.5 Normal approximation to binomial and Poisson	15

	distribution, central limit theorem (statement only), normal probability plot 2.6 Pareto Distribution: p.d.f., mean, variance, applications 2.7 Numerical problems related to real life situations.	
III	**Time Series** 3.1 Meaning and Utility 3.2 Components of Time Series 3.3 Additive and Multiplicative models 3.4 Methods of estimating trend: moving average method, least squares method and exponential smoothing method 3.5 Elimination of trend using additive and multiplicative models 3.6 Measurement and estimation of seasonal variations using link relative method and ratio to trend method, Simple time series models: AR (1), AR (2) 3.7 Numerical problems related to real life situations.	11

Learning Resources:

1. Statistical Methods, G.W. Snedecor, W.G. Cochran, John Wiley & sons, 1989.
2. Fundamentals of Applied Statistics (4th Edition), Gupta and Kapoor, S. Chand and Sons, New Delhi, 2014.
3. Modern Elementary Statistics, Freund J.E., Pearson Publication, 2005.
4. A First course in Probability 6th Edition, Ross, Pearson Publication, 2006.

S.Y B.Sc. (Computer Science) – Semester IV		
STS-266	**Statistics Practical – 4 (Minor Practical)**	**Credits: 02 Hours : 60**
Course Outcomes (COs) On completion of the course, the students will be able to:		**Bloom's cognitive level**
CO1	Identify different real-life situations to find probability of different continuous distributions.	1
CO2	Discuss various applications of statistical measures using R software.	2
CO3	Execute the computational techniques using R software.	3
CO4	Analyze different concepts of statistics using R software.	4
CO5	Validate the fundamental knowledge and represent using R software.	5
CO6	Build models of time series and different methods of estimation of trend and seasonal variation.	6

Sr. No.	Title of Experiment/ Practical
1	Continuous probability theory
2	Applications of Uniform distribution
3	Applications of Exponential distribution
4	Applications of Normal distribution
5	Fitting of Normal distribution
6	Model sampling from continuous probability distributions
7	Computations of probabilities using R-I (Uniform)
8	Computations of probabilities using R-II(Exponential)
9	Computations of probabilities using R-III(Normal)
10	Time Series I (Measurement of trend)
11	Time Series II (Measurement of seasonal variations)
12	Time Series III (Autoregressive models)
13,14&15	Applications of Statistical techniques to real-life data.