SCHEME & SYLLABUS OF
B. Tech. (Internet of Things)

Effective for the students admitted in year 2021-22 and onwards

Approved by ……. academic council meeting held on …….

Teaching & Examination Scheme B.Tech. II Year 3rd Semester

Effective from Session 2021-22

S. No.	Category	Course Code	Course Title	Hours			Marks			Credit
				L	T	P	IA	ETE	Total	
THEORY										
1	UCB	3IO1-01	Advanced Engineering Mathematics	3	-	-	30	70	100	**3**
2	DC	3IO4-02	Digital Electronics	3	-	-	30	70	100	**3**
3	DC	3IO4-03	Data Structures and Algorithms	3	-	-	30	70	100	**3**
4	DC	3IO4-04	Object Oriented Programming Using C++	3	-	-	30	70	100	**3**
5	DC	3IO4-05	Software Engineering	3	-	-	30	70	100	**3**
6	DC	3IO4-06	Introduction to IoT	3	-	-	30	70	100	**3**
Sub Total				**18**	**0**	**0**	**180**	**420**	**600**	**18**
PRACTICAL & SESSIONAL										
7	DC	3IO4-21	Data Structures and Algorithms Lab	-	-	3	60	40	100	**1.5**
8	DC	3IO4-22	Object Oriented Programming Using C++ Lab	-	-	3	60	40	100	**1.5**
9	DC	3IO4-23	Linux and Shell Programming Lab	-	-	2	60	40	100	**1**
10	DC	3IO4-24	Digital Electronics Lab	-	-	2	60	40	100	**1**
11	UI	3IO7-30	Industrial Training (15 Days)	-	-	2	60	40	100	**1**
12	CCA	3IO8-00	SODECA / Co-Curricular Activity	-	-	-	-	100	100	**1**
Sub Total				**0**	**0**	**12**	**300**	**300**	**600**	**7**
Total				**18**	**0**	**12**	**480**	**720**	**1200**	**25**

L = Lecture, **T** = Tutorial, **P** = Practical, **IA** = Internal Assessment, **ETE** = End Term Exam, **Cr** = Credits

Teaching & Examination Scheme B.Tech. 2nd Year – 4th Semester

Effective from Session 2021-22

S. No.	Category	Course Code	Course Title	Hours			Marks			Credit
				L	T	P	IA	ETE	Total	
THEORY										
1	UCB	4IO1-01	Discrete Mathematics	3	-	-	30	70	100	**3**
2	DC	4IO4-02	Microprocessor and Interfaces	3	-	-	30	70	100	**3**
3	DC	4IO4-03	Theory of Computation	3	-	-	30	70	100	**3**
4	DC	4IO4-04	Database Management Systems	3	-	-	30	70	100	**3**
5	DC	4IO4-05	Introduction to Python Programming	3	-	-	30	70	100	**3**
6	DC	4IO4-06	Introduction to Java Programming	3	-		30	70	100	**3**
Sub Total				**18**	**0**	**0**	**180**	**420**	**600**	**18**
PRACTICAL & SESSIONAL										
7	DC	4IO4-21	Database Management Systems Lab	-	-	3	60	40	100	**1.5**
8	DC	4IO4-22	Microprocessor and Interfaces Lab	-	-	3	60	40	100	**1.5**
9	DC	4IO4-23	Python Programming Lab	-	-	3	60	40	100	**1.5**
10	DC	4IO4-24	Java Programming Lab	-	-	3	60	40	100	**1.5**
12	CCA	4IO7-00	SODECA / Co-Curricular Activity	-	-	-	-	100	100	**1**
Sub Total				**0**	**0**	**12**	**240**	**260**	**500**	**7**
Total				**18**	**0**	**12**	**420**	**680**	**1100**	**25**

L = Lecture, **T** = Tutorial, = Practical, **IA**=Internal Assessment**, ETE**=End Term Exam, **Cr**=Credits

III Semester

B. Tech. (Internet of Things)

3IO1-01: Advanced Engineering Mathematics

Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives: This course aims to impart the knowledge of fundamental concepts in probability & statistics, optimization techniques and introduction to the field of numerical analysis.

Course Outcomes: Upon successful completion of the course the students will be able to

CO-1 Compute the discrete and continuous random variables, probability distributions, expectations, moments, MGF, mean and variances.

CO-2 Define and explain the different statistical distributions like Binomial, Poisson, Normal, Uniform, Exponential Distribution and to compute the method of least squares, correlation and regression.

CO-3 To apply the theory of optimization methods to develop and for solving various types of optimization problems.

CO-4 To make aware of the linear programming problem by solving techniques theoretically as well as applications of Linear Programming problem.

CO-5 To study the numerical interpolations for equal and unequal intervals, numerical differentiation, integration and solving ordinary differential equations by numerical methods.

S. No.	Contents	Hours
1	**Probability and Statistics-1:** Discrete and Continuous random variables, Probability distribution function, Mathematical Expectations: Moments, Moment Generating Functions, Mean and variance.	**4**
2	**Probability and Statistics-2:** Binomial distribution, Poisson Distribution, Normal Distribution, Uniform Distribution, Exponential Distribution. Curve fitting, Correlation, Karl Pearson's correlation coefficient, Spearman's Rank correlation coefficient. Lines of Regression, Regression coefficients, Angle between lines of regression	**9**
3	**Optimization Techniques-1:** Historical Development, Engineering applications of Optimization, Single variable Optimization, Multi variable Optimization with and without constraints, Multivariable Optimization with equality constraints--solution by Hessian matrix formulation and method of Lagrange multipliers, Multivariable Optimization with inequality constraints - Kuhn-Tucker conditions.	**8**
4	**Optimization Techniques-2:** Introduction to Linear Programming Problem**,** Simplex method, Big-M Method, Two Phase Method and Duality in Linear Programming. Application of Linear Programming to Transportation and Assignment Problems.	**9**
5	**Numerical Methods**: Finite differences and operators, Interpolation by using Newton's forward and backward difference formula, Gauss's forward and backward interpolation formula, Stirling's formula, Newton's divided difference and Lagrange's interpolation for unequal intervals. Numerical Differentiation. Numerical integration by Trapezoidal rule and Simpson's 1/3 and 3/8 rules. Numerical solution of ordinary differential equations by Euler method, modified Euler's method, Runge- Kutta method of fourth order and Milne's PC methods.	**10**
	Total	**40**

Suggested Books:

- R.K. Jain and S.R.K. Iyengar, Advanced Engineering Mathematics, Fifth Edition, Narosa Publishing House, (2016).
- H.K. Dass, Advanced Engineering Mathematics, 22nd Edition, S. Chand, (2018).
- S.S.Rao, Engineering Optimization: Theory and practice, New Age International (P) Limited, (2009).
- H A Taha, Operations Research: An Introduction, 10th Edition, Pearson Education India, (2017).

- G. Hadley, Linear programming, Narosa Publishing House, New Delhi, (2002).
- Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Academic Press, (2009).
- K. E. Atkinson, An Introduction to Numerical Analysis (2nd edition), Wiley-India, (1989)

III Semester

B. Tech. (Internet of Things)

3IO4-02: Digital Electronics

Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. To present a problem oriented introductory knowledge of Digital circuits and its applications.
2. To focus on the study of electronic circuits.

Course Outcomes: Upon successful completion of the course the students will be able to

CO1: Have a thorough understanding of the fundamental concepts and techniques used in digital electronics.
CO2: To understand and examine the structure of various number systems and its application in digital design.
CO3: The ability to understand, analyze and design various combinational and sequential circuits.
CO4: Ability to identify basic requirements for a design application and propose a cost-effective solution.
CO5: The ability to identify and prevent various hazards and timing problems in a digital design.

S. No.	Contents	Hours
1	**Introduction:** Objective, Scope and Outcome of the course	**1**
2	**Number System, Codes and Logic Gates:** Arithmetic of Nonconventional Number System, Weighted Codes, Binary codes, Code Conversion, Error Correction/Detection Codes, BCD codes, Fixed point & floating-point Number System. Basic, Exclusive and Universal Gates.	**8**
3	**Logic Simplification and Minimization Techniques:** Review of Boolean Algebra and De Morgan's Theorem, SOP & POS forms, Canonical forms, Karnaugh maps up to 6 variables, Tabulation Method.	**7**
4	**Combinational Logic Circuits Design:** Half and Full Adders, Subtractors, Serial and Parallel Adders, BCD Adder, Magnitude Comparators, Multiplexers, Encoder, Decoder, Driver & Multiplexed Display, Logic Implementation using combination blocks.	**8**
5	**Sequential Logic Circuits Design:** Building blocks like S-R, JK and Master-Slave JK FF, Edge triggered FF, Ripple and Synchronous counters, Shift registers, Finite state machines, Design of Synchronous FSM, FSM Minimization, Algorithmic State Machines charts. Designing synchronous circuits like Pulse train generator, Pseudo Random Binary Sequence generator, Clock generation, Asynchronous FSM	**9**
6	**Logic Families:** TTL NAND gate, Specifications, Noise margin, Propagation delay, fan-in, fan-out, Tristate TTL, ECL, CMOS families and their electrical behavior	**7**
	Total	**40**

Suggested Books:

- M. Morris Mano: Digital Design, Third Edition, Prentice Hall
- R. P. Jain: Modern Digital Electronics, Third Edition, TMH
- Taub and Schilling: Digital Integrated Electronics, McGraw HILL
- Sandige: Digital concept Using standard ICs
- R. J. Tocci: Digital Systems: Principles and Applications, Fourth Edition, Prentice Hall
- Z. Kohavi, Switching and Finite Automata Theory, McGraw Hill,1970.

Suggested Books:

- Digital Circuits SWAYAM NPTEL COURSE, By Prof. Santanu Chattopadhyay (IIT Kharagpur), https://onlinecourses.nptel.ac.in/noc19_ee51/preview

III Semester

B. Tech. (Internet of Things)

3IO4-03: Data Structures and Algorithms

Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. To understand the basic concepts of data structures and algorithms.
2. To differentiate linear and non-linear data structures and the operations upon them.
3. Ability to perform sorting and searching in a given set of data items.
4. To comprehend the necessity of time complexity in algorithms.

Course Outcomes: Upon successful completion of the course the students will be able to

CO1: Understanding the fundamental analysis and time complexity for a given problem.
CO2: Articulate linear & non data structures and legal operations permitted on them.
CO3: Applying a suitable algorithm for searching and sorting.
CO4: Understanding graph algorithms, operations, and applications and the importance of hashing.
CO5: Application of appropriate data structures to find solutions to practical problems.

S. No.	Contents	Hours
1	**Introduction to Algorithms and Analysis:** Fundamentals of algorithm analysis, Space and time complexity of an algorithm, Types of asymptotic notations and orders of growth, Algorithm efficiency – best case, worst case, average case, Analysis of non-recursive and recursive algorithms.	**8**
2	**Linear Data Structures:** Array- 1D and 2D array, Stack - Applications of stack: Expression Evaluation - Conversion of Infix to postfix and prefix expression, Tower of Hanoi. Queue - Types of Queues: Circular Queue, Double Ended Queue (deQueue), Applications – Priority Queue using Arrays - List - Singly linked lists – Doubly linked lists - Circular linked lists, Applications -Polynomial Addition/Subtraction	**8**
3	**Sorting and Search Techniques:** Sorting Algorithms: Basic concepts, Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Shell Sort, Heap Sort, Merge Sort, Counting Sort, External Sorting, Internal Sorting, Stable & Unstable Sorting. Searching: Linear Search, Binary Search.	**8**
4	**Trees:** Terminology, Binary Tree – Terminology and Properties, Tree Traversals, Expression Trees – Binary Search Trees – operations in BST – insertion, deletion, Searching. AVL Trees-Insertion, deletion and Rotation in AVL Trees	**7**
5	**Graphs & Hashing:** Basic definition and Terminology – Representation of Graph – Graph Traversal: Breadth First Search (BFS), Depth First Search (DFS) - Minimum Spanning Tree: Prim's, Kruskal's- Single Source Shortest Path: Dijkstra's Algorithm. Hashing: Introduction, open hashing-separate chaining, closed hashing - linear probing, quadratic probing, double hashing, random probing, rehashing, Recent Trends in Data Structures and Algorithms	**9**
	Total	**40**

Suggested Books:

- Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms , Third edition, MIT Press, 2009.
- Ellis Horowitz, S. Sahni, Freed, "Fundamentals of Data Structures in C",2nd edition,2015.
- Y. Langsam, M. J. Augenstein and A. M. Tanenbaum, —Data Structures using C, Pearson Education Asia, 2004.
- Seymour Lipschutz, Data Structures, Schaum's Outlines Series, Tata McGraw-Hill
- Vishal Goyal, Lalit Goyal and Pawan Kumar, Simplified approach to Data Structures, Shroff publications and Distributors.

III Semester B. Tech. (Internet of Things)	
3IO4-04: Object Oriented Programming using C++	
Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. To develop a problem-solving approach using object-oriented programming paradigms.
2. To learn basic concepts and structure syntax of OOP using C++.
3. To learn & implement robust programming using error handling techniques.

Course Outcomes:

Upon successful completion of the course, the students will be able to

CO1: Understand the requirement and benefits of object-oriented programming languages.
CO2: Understand basic concepts & structure of object-oriented programming language using C++.
CO3: Understand the memory management in object-oriented paradigm.
CO4: Understand and implement polymorphism using different ways such as function and operator overloading.
CO5: Learn and implement exception handling mechanism for robust software development in C++.

S. No.	Contents	Hours
1	**Introduction:** Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP. Overview, Program structure, namespace, identifiers, variables, constants, enum, operators, typecasting, control structures, Operators, array and pointer.	**7**
2	**Abstraction mechanism:** Classes, private, public, constructors, destructors, member data, member functions, inline function, friend functions, static members, and references. **Inheritance**: Class hierarchy, derived classes, single inheritance, multiple, multilevel, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors.	**7**
3	**Polymorphism:** Binding, Static binding, Dynamic binding, Static polymorphism: Function Overloading, Ambiguity in function overloading, Dynamic polymorphism: Base class pointer, object slicing, late binding, method overriding with virtual functions, pure virtual functions, abstract classes	**7**
4	**Operator Overloading:** This pointer, applications of this pointer, Operator function, member and nonmember operator function, operator overloading, I/O operators. Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration, unexpected exceptions, exception when handling exceptions, resource capture and release.	**7**
5	**Memory Management:** Dynamic memory management, new and delete operators, object copying, copy constructor, assignment operator, virtual destructor.	**5**
6	**Template:** template classes, template functions. Standard Template Library: Fundamental idea about string, iterators, hashes, iostreams and other types. Namespaces: user defined namespaces, namespaces provided by library. Object Oriented Design, design and programming, role of classes.	**7**
	Total	**40**

Suggested Books:

- Paul Deitel & Harvey Deitel, C++ How to Program, 10th edition, ISBN 9780134448237, Pearson Education
- Robert Lafore, Object Oriented Programming in Turbo C++, Galgotia Publications Pvt Ltd
- Herbert Schlitz, C++: The Complete Reference, McGraw Hill Education India
- Balagurusamy, Object Oriented Programming With C++, 7th Edition, McGraw Hill Education India

III Semester

B. Tech. (Internet of Things)

3IO4-05: Software Engineering

Credit:3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. Provide innovative solutions using technical skills in their discipline
2. Communicate effectively, demonstrate leadership, and work collaboratively in diverse teams/organizations

Course Outcomes: Upon successful completion of the course the students will be able to

CO1: Identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
CO2: Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.
CO3: Communicate effectively with a range of audiences.

S. No.	Contents	Hours
1	**Unit I : Introduction** Introduction, software life-cycle models, software requirements specification, formal requirements specification, verification and validation.	**8**
2	**Unit II: Software Project Management** Software Project Management: Objectives, Resources and their estimation, LOC and FP estimation, effort estimation, COCOMO estimation model, risk analysis, software project scheduling.	**8**
3	**Unit III: Requirement Analysis** Requirement Analysis: Requirement analysis tasks, Analysis principles. Software prototyping and specification data dictionary, Finite State Machine (FSM) models. Structured Analysis: Data and control flow diagrams, control and process specification behavioral modeling	**8**
4	**Unit IV : Software Design** Software Design: Design fundamentals, Effective modular design: Data architectural and procedural design, design documentation.	**8**
5	**Unit V : Object Oriented Analysis** Object Oriented Analysis: Object oriented Analysis Modeling, Data modeling. Object Oriented Design: OOD concepts, Class and object relationships, object modularization, Introduction to Unified Modeling Language	**8**
	Total	**40**

Suggested Books:

- Software Engineering: A Practitioner's Approach by Roger S. Pressman, McGraw-Hill International edition.
- An Integrated Approach to Software Engineering, by Pankaj Jalote, Narosa Publishing House.
- Software Engineering by Ian Sommerville, Addison-Wesley.
- Fundamentals of Software Engineering Rajib Mall, PHI Learning; 5th edition

III Semester B. Tech. (Internet of Things)	
3IO4-06: Introduction to IoT	
Credit:3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

- This Course focuses on hands-on IoT concepts such as sensing, actuation and communication.
- This Course focuses on Internet of Things (IoT) prototypes—including devices for sensing, actuation, processing, and communication—to help you develop skills and experiences.
- This Course focuses on real world applications of IoT.

Course Outcomes: Upon successful completion of the course the students will be able to

- After the completion of the course, the students will be able design some IOT based prototypes.
- The student will able to learn about embedded OS used for IoT application.
- Able to learn about communication protocol of IoT.
- Develop knowledge of IoT hardware and software.
- Develop knowledge of sensors and actuators.
- Study of some real world IoT applications.

S. No.	Contents	Hours
1	**Internet of Things Promises** Definition and characteristics of IoT – Scope–Sensors for IoT Applications–Structure of IoT– IoT Map Device **Design of IOT**: Physical design of IOT, Logical Design of IOT- Functional Blocks, communication models, communication APIs, IOT enabling Technologies- Wireless Sensor Networks, Cloud computing, big data analytics, embedded systems. IOT Levels and deployment templates.	7
2	**IoT Hardware and Software** Arduino, Raspberry Pi, LiteOS, RIoTOS, Contiki OS, Tiny OS, Printed Electronics, IoT Generation Roadmap, Wireless Sensor Structure, Energy Storage Module, Power Management Module, RF Module, Sensing Module	7
3	**IOT and M2M** M2M, Difference and similarities between IOT and M2M, Software defined networks, network function virtualization, difference between SDN and NFV for IoT.	7
4	**Sensor and actuator** Humidity sensors, Ultrasonic sensor, Temperature Sensor, Industrial sensors Description & Characteristics, First Generation, Advanced Generation, Integrated IoT Sensors, Polytronics Systems, Sensors' Swarm	7
5	**Architecture and Reference Model** Introduction, Reference Model and architecture, Representational State Transfer (REST) architectural style, Uniform Resource Identifiers (URIs). Challenges in IoT- Design challenges, Development challenges, Security challenges, Other challenges.	7
6	**Case study of IoT Applications** Domain specific IOTs- Home automation, Cities, environment, Energy, Retail, Logistics, Agriculture, Industry, Health and Lifestyles.	7
	TOTAL	**42**

Suggested Books:

1. Internet of Things : A Hands-on Approach , by Arshdeep Bagha and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547
2. The Internet of Things : Enabling Technologies, Platforms and Use Cases by Pethuru Raj and Anupama

C. Raman (CRC Press)
3. Guillaume Girardin , Antoine Bonnabel, Dr. Eric Mounier, 'Technologies Sensors for the Internet of Things Businesses & Market Trends 2014 -2024',Yole Development Copyrights ,2014
4. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014,
5. Raspberry Pi Cookbook, Software and Hardware Problems and solutions, Simon Monk, O'Reilly (SPD), 2016, ISBN 7989352133895
6. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015 3. Editors Ovidiu Vermesan
7. Peter Friess, 'Internet of Things – From Research and Innovation to Market Deployment', River Publish 2014
8. N. Ida, Sensors, Actuators and Their Interfaces, SciTech Publishers, 2014.

III Semester B. Tech. (Internet of Things)	
3IO4-21: Data Structures and Algorithms Lab	
Credit:1.5	**Max. Marks: 100 (IA:60, ETE:40)**
0L+0T+ 3P	**End Term Exams: 3 Hours**

Course Objectives:

1. To implement an algorithm for a problem and analyze its time and space complexity.
2. To implement the algorithm for Searching (Linear and Binary).
3. To implement the algorithms for the different types of sorting.
4. To implement algorithms for different type of sorting and compare their performance in terms of the space and time complexity

Prerequisites: Computer Programming knowledge.

Course Outcomes: Upon successful completion of the course/Lab the students will be able to

CO1: Be able to design and analyze the time and space efficiency of the data structure.
CO2: Understand the concept of static & Dynamic memory management
CO3: Be capable to identity the appropriate data structure for given problem.
CO4: Have practical knowledge on the applications of data structures

Suggestive List of Experiments

1. Write a program to find the mean and the median of the numbers stored in an array.
2. Write a program to insert one element in an array and delete an element from an- array.
3. Write a program to Linear & Binary search for a number in an array.
4. Write a program to store the marks obtained by 10 students in 5 courses in a two- dimensional array.
5. Write a program to implement single linked list, including insertion, deletion and searching in the linked list.
6. Write a program to print the elements of a linked list in reverse order without disturbing the linked list.
7. Write a program to reverse a linked list.
8. Write a program to add two polynomials using linked lists.
9. Write a program to implement a doubly linked list including insertion, deletion and searching in the linked list.
10. Write a program to implement a stack using an array and linked list.
11. Write a program to implement a queue using an array and linked list.
12. Write a program to implement a circular queue using an array.
13. Write a program to implement a priority queue using a linked list.
14. Write a program to implement a double-ended queue using a linked list.
15. Write a program to implement different types of sorting. (Bubble, Insertion, Quick, Selection, Merge, Heap)
16. Write a program to construct a binary tree and display its preorder, inorder and postorder traversals.
17. Write a program to perform insertion, deletion and searching in Binary Search Tree.
18. Write a program to construct a graph.
19. Write a program to calculate the distance between two vertices in a graph.
20. Write a program to calculate the distances between every pair of vertices in a
21. graph.
22. Write a program to construct a minimal spanning tree of a graph.

Suggested Books:

- Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms , Third edition, MIT Press, 2009.
- Ellis Horowitz, S. Sahni, Freed, “Fundamentals of Data Structures in C”,2nd edition,2015.
- Y. Langsam, M. J. Augenstein & A. M. Tanenbaum, Data Structures using C, Pearson Edu. Asia, 2004.
- Data Structures – Lipshutz TMH

III Semester

B. Tech. (Internet of Things)

3IO4-22: Object Oriented Programming using C++ Lab

Credit: 1.5	Max. Marks: 100 (IA:60, ETE:40)
0L+0T+ 3P	0L+0T+ 3P

Course Objectives:

1. To develop programs in C++ using object-oriented programming paradigms.
2. To design class, object using syntax of C++.
3. To learn & implement all object-oriented mechanism (Encapsulation, Polymorphism, Inheritance, Abstraction) using C++.

Course Outcomes: Upon successful completion of the course/Lab the students will be able to

CO-1: Hands on practice of basic C++ syntax.
CO-2: Hands on practice of class, object and abstraction.
CO3: Hands on practice of inheritance using class hierarchy.
CO4: Hands on practice of function and operator overloading, Templates.
CO5: Hands on practice of exception handling mechanism for robust software development in C++.

Suggestive List of Experiments

Note: Following is a tentative list of experiments covering the syllabus of Object-Oriented Programming using C++. Instructor may add more assignments to the suggested list of experiments covering entire syllabus of Object-Oriented Programming using C++.

1. Write a program that reads in two integers and determines and prints if the first is a multiple of the second.
2. Write a program that reads in the size of the side of a square and then prints a hollow square of that size out of asterisks and blanks. Your program should work for squares of all side sizes between 1 and 20. For example, if your program reads a size of 5, it should print

   *****
   * *
   * *
   * *
   *****
3. Write a program that reads in a five-digit integer and determines whether it is a palindrome.
4. Write a program that computes the value of e^x by using the formula

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

5. Write a program that defines four functions to round a number x in various ways:
   a. roundToInteger(number)
   b. roundToTenths(number)
   c. roundToHundredths(number)
   d. roundToThousandths(number)

   For each value read, your program should print the original value, the number rounded to the nearest integer, the number rounded to the nearest tenth, the number rounded to the nearest hundredth and the number rounded to the nearest thousandth.
6. Write a function ***gcd*** that returns the greatest common divisor of two integers.
7. Write a program to solve the Towers of Hanoi problem. Use a recursive function with four parameters:
   a. The number of disks to be moved
   b. The peg on which these disks are initially threaded
   c. The peg to which this stack of disks is to be moved
   d. The peg to be used as a temporary holding area
8. Write a program that inputs a line of text, tokenizes the line with function **strtok** and outputs the tokens in reverse order. (e.g. for input “Hello dear students” output will be “students” “dear” “Hello”)

9. Create a class called Complex for performing arithmetic with complex numbers. Write a driver program to test your class. Complex numbers have the form ***realPart + imaginaryPart * i***
   Provide public member functions for each of the following:
   a) ***Addition*** of two Complex numbers: The real parts are added together and the imaginary parts are added together.
   b) ***Subtraction*** of two Complex numbers: The real part of the right operand is subtracted from the real part of the left operand and the imaginary part of the right operand is subtracted from the imaginary part of the left operand. c) ***Printing Complex numbers*** in the form (a, b) where a is the real part and b is the imaginary part.
10. Implement overloading of ***operator+*** to allow operations such as ***string1 = string2 + string3***
11. Consider class Complex in problem 9,
    a) Modify the class to enable input and output of complex numbers through the ***overloaded*** >> ***and*** << ***operators***, respectively
    b) ***Overload the multiplication operator*** to enable multiplication of two complex numbers as in algebra.
    c) ***Overload the == and != operators*** to allow comparisons of complex numbers
12. Write a program to develop hierarchy of inheritance for the properties of shapes and their relevant functions.
    e.g. Shapes→2D/3D, 2D→ellipse→circle|rectangle→square and expand it for 3D accordingly.
13. Write a simple function template for predicate function ***isEqualTo*** that compares its two arguments with the equality operator (==) and returns true if they are equal and false if they are not equal. Use this function template in a program that calls ***isEqualTo*** only with a variety of built-in types. Now write a separate version of the program that calls ***isEqualTo*** with a user defined class type, but does not overload the equality operator.
14. Use inheritance to create a base exception class and various derived exception classes. Then show that a catch handler specifying the base class can catch derived-class exceptions.
15. Write a program which shows that all destructors for objects constructed in a block are called before an exception is thrown from that block.
16. Write a program that shows a constructor passing information about constructor failure to an exception handler after a try block.

Suggested Books:

- Paul Deitel & Harvey Deitel, C++ How to Program, 10th edition, ISBN 9780134448237, Pearson Education
- Robert Lafore, Object Oriented Programming in Turbo C++, Galgotia Publications Pvt Ltd
- Herbert Schlitz, C++: The Complete Reference, McGraw Hill Education India
- Balagurusamy, Object Oriented Programming With C++, 7th Edition, McGraw Hill Education India

III Semester B. Tech. (Internet of Things)	
3IO4-23: Linux and Shell Programming Lab	
Credit:1	**Max. Marks: 100 (IA:60, ETE:40)**
0L+0T+ 3P	**End Term Exams: 3 Hours**

Course Objectives:

1. To make familiar with open-source operating system, command line interface, basic commands of Unix/Linux
2. To able to write scripts containing various built-in commands of UNIX/Linux
3. To able to write simple scripts using concepts of control structures of shell programming
4. To able to write basic and advance level of scripts with loops, functions, arrays, etc

Prerequisites: Computer Programming knowledge.

Course Outcomes: Upon successful completion of the course/Lab the students will be able to

CO1: To experiment with various basic commands, redirection and input/output of UNIX based operating systems.

CO2: To develop shell scripts for various built-in commands of UNIX

CO3: To experiment with fundamental concepts of programming like loops, conditions, operators etc specific to Shell Programming.

CO4: To develop shell scripts to perform tasks varying from simple to complex level.

Suggestive List of Experiments

1. Use of Basic Unix Shell Commands: ls, mkdir, rmdir, cd, cat, banner, touch, file, wc, sort, cut, grep, dd, dfspace, du, ulimit.
2. Commands related to inode, I/O redirection and piping, process control commands, mails.
3. Shell Programming: Shell script based on control structure- If-then-fi, if-thenelse-if, nested if-else, to find:
    - 3.1 Greatest among three numbers.
    - 3.2 To find a year is leap year or not.
    - 3.3 To input angles of a triangle and find out whether it is valid triangle or not.
    - 3.4 To check whether a character is alphabet, digit or special character.
    - 3.5 To calculate profit or loss.
4. Shell Programming - Looping- while, until, for loops
    - 4.1 Write a shell script to print all even and odd number from 1 to 10.
    - 4.2 Write a shell script to print table of a given number
    - 4.3 Write a shell script to calculate factorial of a given number.
    - 4.4 Write a shell script to print sum of all even numbers from 1 to 10.
    - 4.5 Write a shell script to print sum of digit of any number.
5. Shell Programming - case structure, use of break
    - 5.1 Write a shell script to make a basic calculator which performs addition, subtraction, Multiplication, division
    - 5.2 Write a shell script to print days of a week.
    - 5.3 Write a shell script to print starting 4 months having 31 days.
6. Shell Programming - Functions
    - 6.1 Write a shell script to find a number is Armstrong or not.
    - 6.2 Write a shell script to find a number is palindrome or not.
    - 6.3 Write a shell script to print Fibonacci series.
    - 6.4 Write a shell script to find prime number.
    - 6.5 Write a shell script to convert binary to decimal and decimal to binary
7. Write a shell script to print different shapes- Diamond, triangle, square, rectangle, hollow square etc.
8. Shell Programming – Arrays
    - 8.1 Write a Shell script to read and print elements of array.
    - 8.2 Write a Shell script to find sum of all array elements.
    - 8.3 Write a Shell script to find reverse of an array.
    - 8.4 Write a Shell script to search an element in an array.

8.5 Write a Shell script to sort array elements in ascending or descending order.

Suggested Books:

- Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms , Third edition, MIT Press, 2009.
- Ellis Horowitz, S. Sahni, Freed, "Fundamentals of Data Structures in C",2nd edition,2015.
- Y. Langsam, M. J. Augenstein and A. M. Tanenbaum, ―Data Structures using C, Pearson Education Asia, 2004.
- Data Structures – Lipshutz TMH

III Semester

B. Tech. (Internet of Things)

3IO4-24: Digital Electronics Lab

Credit:1	**Max. Marks: 100 (IA:60, ETE:40)**
0L+0T+ 3P	**End Term Exams: 3 Hours**

Course Objectives:
To present a problem oriented introductory knowledge of Digital circuits and its applications.
To focus on the study of electronic circuits.

Course Outcomes: Upon successful completion of the course/Lab the students will be able to
CO1: Understand different Number systems, Codes, Logic Gates, Boolean laws & theorems.
CO2: Simplify the Boolean functions to the minimum number of literals.
CO3: Design & implement different types of combinational logic circuits using Logic gates.
CO4: Design & implement different types of sequential logic circuits using Flip Flops.
CO5: Design & implement different types of Counters, Registers, and Programmable Logic Devices.

Suggestive List of Experiments

1. Realization of Basic/ Exclusive Logic Gates using Universal Logic Gate.
2. Verification of operation of Full Adder and Full Subtractor.
3. Design & verification of 4-bit binary adder/subtractor using binary adder IC.
4. Realization of operation of full adder and full subtractor using IC 74151/74153 MUX.
5. Design & verification of full adder and full subtractor using an inverted output 3 to 8 line decoder.
6. Design and verification of operation of a BCD Adder using IC 7483.
7. Realization of 4 X 1 MUX using basic gates.
8. Verification of operation of BCD to Seven segment code conversion using IC 7447.
9. Verification of Truth Tables of SR & D Flip flops.
10. Verification of Truth Tables of Master Slave JK Flip-Flop.
11. Design of BCD ripple counter.
12. Design of Universal Shift Register.
13. Logic implementation using programmable Devices (ROM , PLA, FPGA)

Suggested Books:

- M. Morris Mano: Digital Design, Third Edition, Prentice Hall
- R. P. Jain: Modern Digital Electronics, Third Edition, TMH
- Taub and Schilling: Digital Integrated Electronics, McGraw HILL
- Sandige: Digital concept Using standard ICs
- R. J. Tocci: Digital Systems: Principles and Applications, Fourth Edition, Prentice Hall
- Z. Kohavi, Switching and Finite Automata Theory, McGraw Hill,1970.

IV Semester
B. Tech. (Internet of Things)

4IO1-01: Discrete Mathematics

Credit:3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives

1. To understand the concepts of mathematical logic, sets, relations and functions.
2. To understand generating functions and recurrence relations.
3. To understand combinatorial mathematics.
4. To identify the basic properties of graphs and trees.

Course Outcomes: Upon successful completion of the course the students will be able to

CO-1: Understand the language of logic.
CO-2: Understand the concept of sets, relations, functions and counting principles.
CO-3: Understand different terminologies and theorems of Graph Theory.
CO-4: Understand Algebraic Structures.

S. No.	Contents	Hours
1	**Propositional Logic**: Propositions and compound Propositions, Basic logical operations, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, Normal forms, predicates and quantifiers, Rules of Inference. **Theorem proving Techniques:** Mathematical induction, Introduction to Proofs, Methods of proof.	**6**
2	**Set Theory**: Definition of sets, countable and uncountable sets, Set operations, Partition of set, Cardinality (Inclusion-Exclusion & Addition Principles) Venn Diagrams, proofs of some general identities on sets. **Relation**: Definition, types of relation, composition of relations, Equivalence relation, Partial ordering relation. **Function**: Definition, type of functions, one to one, into and onto function, inverse function, composition of functions, recursively defined functions. **Posets, Hasse Diagram and Lattices**: Introduction, ordered set, Hasse diagrams of partially ordered set, isomorphic ordered set, well ordered set, properties of lattices, bounded and complemented lattices.	**8**
3	**Combinatorics**: The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients and Identities. **Recurrence Relation and Generating Function**: Introduction to Recurrence Relation and Recursive algorithms, linear recurrence relations with constant coefficients, Homogeneous solution, Particular solution, Total solution, Generating functions, Solution by method of generating functions.	**8**
4	**Graph Theory**: Graphs and Multi-graphs, Degree of a vertex, Paths connectivity, Walks, Paths, Cycles, Bipartite, Regular, Planar and connected graphs, Components, Euler graphs, Euler's theorem, Hamiltonian path and circuits, Graph coloring, chromatic number, isomorphism and homomorphism of graphs. Trees, properties of trees, pendant vertices in trees, Degree sequences in trees, Rooted and Binary Trees, Minimal Spanning Trees.	**10**
5	**Algebraic Structures**: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, Properties of groups, Subgroup, cyclic group, Permutation group, Cosets, Normal subgroup, Quotient Group, Homomorphism and isomorphism of Groups, example and standard results.	**8**
	Total	**40**

Suggested Books:

- Kenneth H. Rosen, Discrete Mathematics and its applications, 7th Ed. Tata McGraw Hill (2012).
- C. L. Liu, Elements of Discrete Mathematics, 2nd Ed., Tata Mc-Graw Hill (2005) .
- Kolman, Busby and Ross, Discrete Mathematical Structures, 6th Ed. PHI (2009).

- Narsingh Deo, Graph Theory with Applications to Engineering and Computer Sciences, PHI (2020).
- Murry R. Spiegel, Discrete Mathematics (Schaums Outline series), Tata McGraw Hill (2009).
- I.N. Herstein, Topics in Algebra, Wiley (2022).

IV Semester

B. Tech. (Internet of Things)

4IO4-02: Microprocessor and Interfaces

Credit:3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. Demonstrate the various features of microprocessor, memory and I/O devices including concepts of system bus.
2. Identify the hardware elements of 8085 microprocessor including architecture and pin functions and programming model including registers, instruction set and addressing modes.
3. Select appropriate 8085 instructions based on size and functions to write a given assembly language program.
4. Design a given interfacing system using concepts of memory and I/O interfacing.
5. Demonstrate the features of advance microprocessors.

Course Outcomes: Upon successful completion of the course the students will be able to

CO1: Basic understanding of 8085 microprocessor, timing diagram and memory mapping.
CO2: Understand ISA for 8085 and also How to design ISA for some other microprocessors.
CO3: Write basic program in assembly language and concept of other Programmable peripheral devices.
CO4: Interface I/O devices, interrupt controller and DMA.
CO5: Basic understanding of design ISA and further design their own processor.

S. No.	Contents	Hours
1	**Introduction:** Objective, Scope and Outcome of the course	**1**
2	**Introduction and architecture of 8085:** Microprocessor Architecture &Operations, Memory, I/O Device, Memory and I/O Operations, , Address, Data And Control Buses, Pin Functions, concept of multiplexing and de-multiplexing of buses, Generation Of Control Signals, Instruction Cycle, Machine Cycles, T-States, Memory Interfacing.	**7**
3	**Instruction set and assembly language programming:** Introduction to 8085 assembly language programming, Instruction Set, Addressing modes, Data transfer, arithmetic, logical, branch, stack and machine control groups of instruction set, macro RTL and micro RTL flow chart of instructions, Code Conversion, BCD Arithmetic and 16-Bit Data operations	**8**
4	**Interfacing with I/O Devices:** Interfacing Concepts, Ports, Interfacing of I/O Devices, Interrupts in 8085, Programmable Interrupt Controller 8259A, Programmable Peripheral Interface 8255A, 8257 (DMA Controller), 8253/8254 (Programmable Interval Timer).	**8**
5	**Introduction and architecture of 8051 Microcontroller:** Internal Block Diagram, CPU, ALU, address, data and control bus, Working registers, SFRs, Clock and RESET circuits, Stack and Stack Pointer, Program Counter, I/O ports, Memory Structures, Data and Program Memory, Timing diagrams and Execution Cycles.	**8**
6	**Programming and application of 8051 Microcontroller:** Programming Timer interrupts, programming external hardware interrupts, Programming the serial communication interrupts, Programming 8051 timers and counters.	**8**
	Total	**40**

Suggested Books:

- Hall D.V.,"Microprocessor and Interfacing-Programming and Hardware", 2nd Ed., Tata McGraw-Hill Publishing Company Limited, 2008.
- GaonkarR.S.,"Microprocessor Architecture ,Programming and Applications", 5th Ed., Penram International, 2007.
- Stewart J,"Microprocessor Systems- Hardware,Software and Programming", Prentice Hall International Edition,1990

IV Semester

B. Tech. (Internet of Things)

4IO4-03: Theory of Computation

Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. Understand the relationship between languages, grammars and automaton models.
2. Design automation for different strings or machine
3. To study the capabilities of the abstract machines.
4. Understanding the theoretical limits of computation and identify the NP complete and NP Hard problems
5. Classify machines by their power to recognize languages.

Course Outcomes:

Upon successful completion of the course the students will be able to

CO-1: Able to classify Language and Grammar in Type0, Type1, Type2 and Type3. Design the Grammar for given string or languages.

CO-2: Able to design the FA, PDA and TM for given string and languages.

CO-3: Able to convert PDA to CFG. Able to apply the pumping lemma for regular languages

CO-4: Able to demonstrate that a grammar is ambiguous. Simplification of the CFG, representations of grammars in CNF and GNF.

CO-5: Understanding the concepts of LBA, NP Complete and NP Hard.

S. No.	Contents	Hours
1	**Finite Automata & Regular Expression**: Basic machine, Finite state machine, Transition graph, Transition matrix, Deterministic and nondeterministic finite automation, Equivalence of DFA and NDFA, Decision properties, minimization of finite automata, Mealy & Moore machines. Alphabet, words, Operations, Regular sets, relationship and conversion between Finite automata and regular expression and vice versa, designing regular expressions, closure properties of regular sets, Pumping lemma and regular sets, Myhill- Nerode theorem, Application of pumping lemma, Power of the languages.	**8**
2	**Context Free Grammars:** CFG, Derivations and Languages, Relationship between derivation and derivation trees, leftmost and rightmost derivation, sentential forms, parsing and ambiguity, simplification of CFG, normal forms, Greibach and Chomsky Normal form, Problems related to CNF and GNF including membership problem.	**8**
3	**PushDown Automaton:** Nondeterministic PDA, Definitions, PDA and CFL, CFG for PDA, Deterministic PDA, and Deterministic PDA and Deterministic CFL , The pumping lemma for CFL's, Closure Properties and Decision properties for CFL, Deciding properties of CFL.	**8**
4	**Turing Machines:** Introduction, Definition of Turing Machine, TM as language Acceptors and Transducers, Computable Languages and functions, Universal TM & Other modification, multiple tracks Turing Machine. Hierarchy of Formal languages: Recursive & recursively enumerable languages, Properties of RL and REL, Introduction of Context sensitive grammars and languages, The Chomsky Hierarchy.	**8**
5	**Tractable and Un-tractable Problems:** P, NP, NP complete and NP hard problems, Un-decidability, examples of these problems like vertex cover problem, Hamiltonian path problem, traveling sales man problem.	**8**
	Total	**40**

Suggested Books:

- K L P Mishra and N Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, Prentice Hall India Learning Private Limited
- John C. Martin, Introduction to Languages and The Theory of Computation, McGraw-Hill
- Aho, Hopcroft and Ullman, Introduction to Automata Theory, Formal Languages and Computation, Narosa
- Cohen, Introduction to Computer Theory, Addison Wesley.

IV Semester B. Tech. (Internet of Things)	
4IO4-04: Database Management Systems	
Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L+0T+ 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. To understand purpose of database management systems.
2. Apply concepts of database design and database languages (SQL based) in managing data.
3. Understand concepts and importance of relational algebra and relational calculus.
4. Importance and application of normalization in DBMS.
5. Knowledge of transaction, concurrency control, recovery strategies.

Course Outcomes:

Upon successful completion of the course the students will be able to

CO-1: Describe DBMS architecture, physical and logical database designs, database models, entity-relationship model.

CO-2: Understand relational algebra, relational calculus importance and query writing

CO-3: Apply Structured query language (SQL) for database definition, database manipulation, data control.

CO-4: Understanding of normalization theory and apply it to normalize databases.

CO-5: Understand various transaction processing, concurrency control mechanisms and database protection mechanisms.

S. No.	Contents	Hours
1	**Introduction to database systems:** Overview and History of DBMS. File System v/s DBMS. Advantage of DBMS Describing and Storing Data in a DBMS. Queries in DBMS. Structure of a DBMS. **Entity Relationship model:** Overview of Data Design Entities, Attributes and Entity Sets, Relationship and Relationship Sets. Features of the ER Model- Key Constraints, Participation Constraints, Weak Entities, Class Hierarchies, Aggregation, Conceptual Data Base, and Design with ER Model- Entity v/s Attribute, Entity vs Relationship Binary vs Ternary Relationship and Aggregation v/s ternary Relationship Conceptual Design for a Large Enterprise.	**8**
2	**Relationship Algebra and Calculus:** Relationship Algebra Selection and Projection, Set Operations, Renaming, Joints, Division, Relation Calculus, Expressive Power of Algebra and Calculus. SQL queries programming and Triggers: The Forms of a Basic SQL Query, Union, and Intersection and Except, Nested Queries, Correlated Nested Queries, Set-Comparison Operations, Aggregate Operators, Null Values and Embedded SQL, Dynamic SQL, ODBC and JDBC, Triggers and Active Databases.	**8**
3	**Schema refinement and Normal forms:** Introductions to Schema Refinement, Functional Dependencies, Boyce-Codd Normal Forms, Third Normal Form, Normalization-Decomposition into BCNF Decomposition into 3-NF.	**8**
4	**Transaction Processing:** Introduction-Transaction State, Transaction properties, Concurrent Executions. Need of Serializability, Conflict vs. View Serializability, Testing for Serializability, Recoverable Schedules, Cascadeless Schedules.	**8**
5	**Concurrency Control:** Implementation of Concurrency: Lock-based protocols, Timestamp-based protocols, Validation-based protocols, Deadlock handling, **Database Failure and Recovery:** Database Failures, Recovery Schemes: Shadow Paging and Log-based Recovery, Recovery with Concurrent transactions.	**8**
Total		**40**

Suggested Books:

- H. F. Korth and Silberschatz: Database Systems Concepts, McGraw Hill
- Almasri and S. B. Navathe: Fundamentals of DataBase Systems
- Ramakrishnan: Database Management Systems
- C. J. Date: Data Base Design, Addison Wesley
- Hansen and Henson: DBM and Design, PHI

IV Semester B. Tech. (Internet of Things)	
4IO4-05: Introduction to Python Programming	
Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L + 0T + 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. Develop understanding of the fundamental concepts essential for programming.
2. To enable students to design algorithms, apply code and data visualized the data.
3. To enable students to apply python programming in problem solving.

Course Outcomes:

Upon successful completion of the course the students will be able to

CO-1: Know the Essential concepts of Python Programming and its real time use.

CO-2: Design algorithms and source code.

CO-3: Use of suitable data structure and logic for problem solving.

S. No.	Contents	Hours
1	**Introduction to Python:** Why Python? - Essential Python libraries - Python Introduction- Features, Data types, variables, expressions, operators, Identifiers, Reserved words, Indentation, Comments.	**8**
2	**Decision Making**: Selective statements – if, if-else, nested if, if –elif ladder statements. Iterative statements - while, for, Nested loops, else in loops, break, continue and pass statements. **Looping**: Loop Control statement- Math and Random number functions. User-defined functions - function arguments & its types. **Strings**: Formatting, Comparison, Slicing, Splitting, Stripping, Negative indices, String functions. **Regular expression**: Matching the patterns, Search and replace.	**8**
3	**List**: Create, Access, Slicing, Negative Indices, List Methods, and comprehensions. **Tuples**: Create, Indexing and Slicing, Operations on tuples. **Dictionary**: Create, add, and replace values, operations on dictionaries. **Sets**: Create and operations on set.	**8**
4	**Functions**: Types, parameters, arguments: positional arguments, keyword arguments, parameters with default values, functions with arbitrary arguments, Scope of variables: Local and global scope, Recursion and Lambda functions. **Files**: Open, Read, Write, Append and Close. Tell and seek methods	**8**
5	**NumPy Basics**: Arrays and Vectorized Computation- The NumPy ND array- Creating ND arrays- Data Types for ND arrays- Arithmetic with NumPy Arrays- Basic Indexing and Slicing- Boolean Indexing-Transposing Arrays and Swapping Axes. Universal Functions: Fast Element-Wise Array Functions- Mathematical and Statistical Methods-Sorting Unique and Other Set Logic. , Data Visualization	**8**
Total		**40**

Suggested Books:

- Programming Python by Mark Lutz, O'Reilly.
- Learning Python, 3rd Edition by Mark Lutz, O'Reilly
- Python in a Nutshell by Alex Martelli, O'Reilly.
- Wesley J. Chun, "Core Python Programming", Prentice Hall,2006.
- Mark Lutz, "Learning Python", O'Reilly, 4th Edition, 2009.
- Introduction to Programming using Python by Y. Daniel Liang , Pearson,2012.

IV Semester B. Tech. (Internet of Things)	
4IO4-06: Introduction to Java Programming	
Credit: 3	**Max. Marks: 100 (IA:30, ETE:70)**
3L + 0T + 0P	**End Term Exams: 3 Hours**

Course Objectives:

1. To understand the basic concepts and fundamentals of platform independent object-oriented language
2. To demonstrate skills in writing programs using exception handling techniques and multithreading.
3. To understand streams and efficient user interface design techniques.

Course Outcomes:

Upon successful completion of the course the students will be able to

CO-1: Understand the features of Java such as operators, classes, objects, inheritance, packages and exception handling

CO-2: Learn latest features of Java like garbage collection, Console class, Network interface, APIs

CO-3: Acquire competence in Java through the use of multithreading, applets

CO-4: Get exposure to advance concepts like socket and database connectivity.

S. No.	Contents	Hours
1	**Introduction**: Object oriented programming principles, Java essentials, java virtual machine, program structure in java, Java class libraries, Data types, Variables and Arrays, Data types and casting, automatic type promotion in expressions, arrays. Operators and Control Statements: Arithmetic operators, bit wise operators, relational operators, Boolean logical operators, the ? Operator, operator precedence, Java's selection statements, iteration statements, jump statements.	**8**
2	**Introduction to Classes**: Class fundamentals, declaring class, creating objects, introducing methods: method declaration, overloading, using objects as parameters, recursion, Constructors, this keyword, garbage collection, the finalization.	**6**
3	**Inheritance:** Inheritance basics, using super and final, method overriding, dynamic method dispatch, Abstract Class, Interface: variables and extending Interfaces, Package: Creating and importing packages, Package access protection, Exception Handling: Exception handling fundamentals, Exception types, Uncaught Exceptions Using try and catch, multiple catch clauses, nested try statements, throw, Java's built-in exceptions.	**10**
4	**Multithreaded Programming:** The Java thread model, the main thread, creating thread, creating multiple threads, using isAlive () and join (), Thread priorities, synchronization, inter thread communications, suspending resuming and stopping threads.	**10**
5	**I/O Operations:** I/O Basics, Reading Console Input, Writing Console Output, Reading and Writing Files , Applets: Applet Fundamentals, Applet Architecture, The HTML Applet tag, Passing parameters to Applets., Networking: Networking basics, Java and the Net, TCP/IP Client Sockets URL, URL Connection, TCP/IP Server Sockets, Database connectivity.	**6**
Total		**40**

Suggested Books:

- Herbert Schildt, The Complete Reference Java 2, McGraw-Hill.
- Joyce Farrell, Java for Beginners, Cengage Learning.
- Deitel and Deitel, Java: How to Program, 6th Edition, Pearson Education.
- James Edward Keogh, Jim Keogh, J2EE: The complete Reference, McGrawHill
- Khalid A. Mughal, Torill Hamre, Rolf W. Rasmussen, Java Actually, Cengage Learning.
- Shirish Chavan, Java for Beginners, 2nd Edition, Shroff Publishers.

IV Semester B. Tech. (Internet of Things)	
4IO4-21: Database Management Systems Lab	
Credit: 1.5	**Max. Marks: 100 (IA:60, ETE:40)**
0L+ 0T+ 3P	**End Term Exams: 3 Hours**

Course Objectives:

1. Installing and configuring databases such as MySQL on windows and Linux platforms along with front end tools.
2. Designing database for different applications and applying various DDL queries along with various Integrity constraints.
3. Write various DML queries using select clause with join, subqueries, group operations etc.
4. Creating triggers and views. Writing DCL queries for creating users, rights.
5. Design and Implement database including E-R model and Relational model for one application like college management, Hospital management along with front end.

Course Outcomes: Upon successful completion of the course/Lab the students will be able to

CO1: Installation of Backend and front end.
CO2: Writing DDL queries effectively.
CO3: Writing advance DML queries in MySQL.
CO4: Writing DCL queries, triggers and views.
CO5: Developing a web-based or client server-based application.

Suggestive List of Experiments

1. Design a Database and create required tables. For e.g. Bank, College Database
2. Apply the constraints like Primary Key, Foreign key, NOT NULL to the tables.
3. Write a SQL statement for implementing ALTER, UPDATE and DELETE.
4. Write the queries to implement the joins.
5. Write the query for implementing the following functions: MAX (), MIN (), AVG () and COUNT ().
6. Write the query to implement the concept of Integrity constrains.
7. Write the query to create the views.
8. Perform the queries for triggers.
9. Perform the following operation for demonstrating the insertion , updation and deletion
10. Using the referential integrity constraints.
11. Write the query for creating the users and their role.
12. **Data Base Designing Project**: For better understanding students (group of 3-4 students) should design web based project containing data base, understand the requirements and design the front end and backend of project by its own. Some example of data base design project like: College management system, Inventory management system and Hospital management system.

Suggested Books:

- Hall D. V., "Microprocessor and Interfacing-Programming and Hardware", 2nd Ed., Tata McGraw-Hill Publishing Company Limited, 2008.
- Gaonkar R. S., "Microprocessor Architecture, Programming and Applications", 5th Ed., Penram International Publishing, 2007.
- Stewart J, "Microprocessor Systems- Hardware, Software and Programming", Prentice Hall International Edition,1990

IV Semester B. Tech. (Internet of Things)	
4IO4-22: Microprocessor and Interfaces Lab	
Credit: 1.5	**Max. Marks: 100 (IA:60, ETE:40)**
0L+ 0T+ 3P	**End Term Exams: 3 Hours**

Course Objectives:

1. Demonstrate the various features of microprocessor, memory and I/O devices including concepts of system bus.
2. Identify the hardware elements of 8085 microprocessor including architecture and pin functions and programming model including registers, instruction set and addressing modes.
3. Select appropriate 8085 instructions based on size and functions to write a given assembly language program.
4. Design a given interfacing system using concepts of memory and I/O interfacing.
5. Demonstrate the features of advance microprocessors.

Course Outcomes: Upon successful completion of the course/Lab the students will be able to

CO1: Ability to write assembly language program for data transfer and control instructions.
CO2: Ability to write assembly language program for Arithmetic calculation using register pair.
CO3: Ability to Write assembly language program for interfacing with Programmable peripheral devices.
CO4: Assembly language programming for general purpose problems like traffic light controller, control the speed of step motor etc.
CO5: To make live projects using assembly language and interfacing with PPI and see outputs on CRO and other electronic devices.

Suggestive List of Experiments

1. Study the hardware, functions, memory structure, Instruction set and operation of 8085 microprocessor kit.
2. Write an assembly language program to Add/Subtract two 8-bit/16-bit number.
3. Write an assembly language program to Data transfer/Exchange from one memory block to another in forward and reverse order.
4. Write an assembly language program to generate a square wave of 1khz frequency on the SOD pin of 8085. Operating frequency of 8085 is 3 khz.
5. Write an assembly language program to perform following conversion:
   (i) BCD to ASCII
   (ii) BCD to Hexadecimal.
6. Write an assembly language program for Sorting of array(Ascending/Descending), Searching a number in array, find largest/smallest number in array and to generate Fibonacci series.
7. Design your own minimal set of ISA similar to 8085 which will compute all arithmetic and logic, memory and control instruction(you have to introduce addressing mode in ISA)
   Design microprocessor using above minimal set of ISA(experiment number 7) which will perform all computation and implement using FPGA

Suggested Books:

- Hall D. V., "Microprocessor and Interfacing-Programming and Hardware", 2nd Ed., Tata McGraw-Hill Publishing Company Limited, 2008.
- Gaonkar R. S., "Microprocessor Architecture, Programming and Applications", 5th Ed., Penram International Publishing, 2007.
- Stewart J, "Microprocessor Systems- Hardware, Software and Programming", Prentice Hall International Edition,1990

IV Semester B. Tech. (Internet of Things)	
4IO4-23: Python Programming Lab	
Credit: 1.5	**Max. Marks: 100 (IA:60, ETE:40)**
0L+ 0T+ 3P	**End Term Exams: 3 Hours**

Course Objectives:

1. To provide skills for designing algorithm and writing code.
2. To introduce students to the real word programming applications using Python.
3. To enable students to apply data structure & python programming code in problem solving.

Course Outcomes:

Upon successful completion of the course the students will be able to

CO-1: Demonstrate and understanding of programming language concepts.
CO-2: Identify and abstract the programming task involved for a given problem.
CO-3: Design and develop modular programming skills.
CO-4: Trace and debug a program.

S. No.	Contents
1	Installation of Python, and learning interactively at command prompt and writing simple programs.
2	Perform Creation, indexing, slicing, concatenation, and repetition operations on Python built-in data types: Strings, List, Tuples , Dictionary, Set
3	Solve problems using decision and looping statements
4	Handle numerical operations using math and random number functions
5	Create user-defined functions with different types of function arguments.
6	Perform File manipulations- open, close, read, write, append and copy from one file to another.
7	Matrix addition, multiplications, and unity matrix.
8	Text processing using python , Import a CSV file and perform various Statistical and Comparison operations on rows/columns.
9	Intrinsic NumPy objects and Random Functions. Manipulation of NumPy arrays- Indexing, Slicing, Reshaping, Joining, and Splitting.
10	Programs related to python libraries like Numpy, Pandas, Scipy etc.

Suggested Books:

- Beginning Python Wrox Publication Peter Norton, Alex Samuel
- Starting Out with Python (2009) Pearson, Tonny Gaddis
- Y. Daniel Liang, "Introduction to Programming using Python," Pearson,2012.
- Wes McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython," O'Reilly, 2nd Edition,2018.
- Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data," O'Reilly, 2017.

IV Semester B. Tech. (Internet of Things)	
4IO4-24: Java Programming Lab	
Credit: 1.5	**Max. Marks: 100 (IA:60, ETE:40)**
0L+ 0T+ 3P	**End Term Exams: 3 Hours**

Course Objectives:

1. To write programs using abstract classes
2. To write multithreaded programs.
3. To write GUI programs in Java.
4. To impart hands on experience with java programming.

Course Outcomes:

Upon successful completion of the course the students will be able to

CO-1: Implement the features of Java such as operators, classes, objects, inheritance, packages and exception handling

CO-2: Design problems using latest features of Java like garbage collection, Console class, Network interface, APIs

CO-3: Develop competence in Java through the use of multithreading, Applets etc

CO-4: Apply advance concepts like socket and database connectivity, and develop project based on industry orientation

S. No.	Contents
1	WAP in Java to show implementation of classes.
2	WAP in Java to show implementation of inheritance.
3	WAP in Java to show Implementation of packages and interfaces. To accomplish
4	WAP in Java to show Implementation of threads.
5	WAP in Java Using exception handling mechanisms.
6	WAP in Java to show Implementation of Applets.
7	WAP in Java to show Implementation of mouse events, and keyboard events.
8	WAP in Java to show Implementing basic file reading and writing methods.
9	Using basic networking features, WAP in Java To accomplish
10	WAP in Java to show Connecting to Database using JDBC.
11	Project work: A desktop based application project should be designed and implemented in java.

Suggested Books:

- Herbert Schildt, The Complete Reference Java2, McGraw-Hill. 2.
- Deitel and Deitel, Java: How to Program, 6th Edition, Pearson Education.
- James Edward Keogh, Jim Keogh, J2EE: The complete Reference, McGrawHill