Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering

Sector No. 26, Pradhikaran, Nigdi, Pune 411044

An Autonomous Institute Approved by AICTE and affiliated to SPPU, Pune

Curriculum Structure and Syllabus

B. Tech. Information Technology

Approved by BoS IT on 22.07.21

(Course 2020)

Department of Information Technology

Effective from Academic Year 2020-21

VISION AND MISSION OF INSTITUTE

Institute Vision

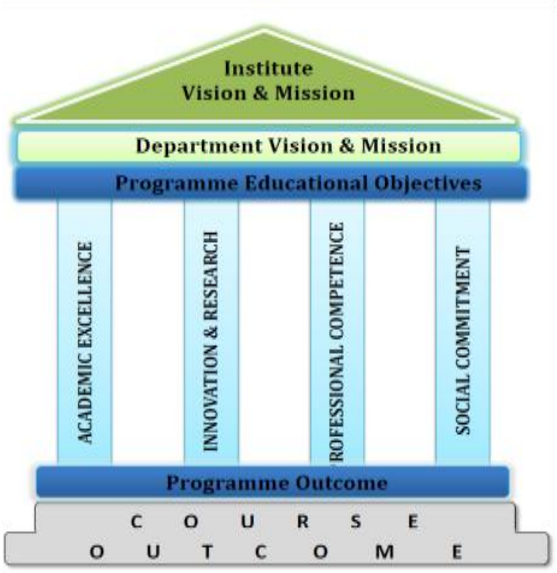
To Serve the Society, Industry and all the Stakeholders through the **Value-Added Quality Education.**

Institute Mission

To serve the needs of society at large by establishing State-of-the-Art Engineering, Management and Research Institute and impart attitude, knowledge and skills with quality education to develop individuals and teams with ability to think and analyze right values and self-reliance.

Quality Policy

We at PCCOE are committed to impart Value Added Quality Education to satisfy the applicable requirements, needs and expectations of the Students and Stakeholders. We shall strive for academic excellence, professional competence and social commitment in fine blend with innovation and research. We shall achieve this by establishing and strengthening state-of-the-art Engineering and Management Institute through continual improvement in effective implementation of Quality Management System.



INDEX

Sr. No.	Content	Pg. No.
1.	Curriculum Framework	1
2.	Structure of S.Y. B.Tech IT	3
3.	Curriculum Syllabus (Semester-III)	5
4.	Curriculum Syllabus (Semester-IV)	21

Curriculum Framework for B.Tech IT

Sr. No.	Type of course	Abbreviations
1	Basic Science Course	**BSC**
2	Engineering Core/Science Course	**ECC**
3	Humanities, Social Sciences and Management Course	**HSMC**
4	Professional Core Course	**PCC**
5	Professional Elective Course	**PEC**
6	Open Elective Course	**OEC**
7	Project	**PROJ**
8	Internship	**INTR**
9	Audit Course	**AC**
10	Mandatory Course	**MC**
11	Life Skills	**LS**
12	Proficiency Course	**PFC**
13	MOOC	**MO**

Sr. No.	Type of course	No. of Courses	Total Credits	
			No	%
1	Basic Science Course	**8**	**23**	**14.3**
2	Engineering Core/Science Course	**14**	**22**	**13.7**
3	Humanities, Social Sciences and Management Course	**6**	**13**	**8.1**
4	Professional Core Course	**22**	**48**	**29.8**
5	Professional Elective Course	**6**	**18**	**11.2**
6	Open Elective Course	**6**	**18**	**11.2**
7	Project	**2**	**16**	**9.9**
8	Internship	**1**	**3**	**1.9**
9	Audit Course	**3**	**0**	**0.0**
10	Mandatory Course	**2**	**0**	**0.0**
11	Life Skills	**4**	**0**	**0.0**
12	Proficiency Course	**3**	**0**	**0.0**
13	MOOC	**1**	**0**	**0.0**
	Total	**78**	**161**	**100.0**

Sr. No.	Type of course	No. of Courses/Semester								Total
		1	2	3	4	5	6	7	8	
1	Basic Science Course	3	3	2						**8**
2	Engineering Core/Science Course	5	6	2	1					**14**
3	Humanities, Social Sciences and Management Course	1	1	1	1	1	1			**6**
4	Professional Core Course			5	6	3	4	4		**22**
5	Professional Elective Course					2	2	2		**6**
6	Open Elective Course				1	1	2	2		**6**
7	Project	1							1	**2**
8	Internship								1	**1**
9	Audit Course				1	1	1			**3**
10	Mandatory Course					1	1			**2**
11	Life Skills	1	1	1	1					**4**
12	Proficiency Course				1	1	1			**3**
13	MOOC								1	**1**
Total		**11**	**11**	**11**	**12**	**10**	**12**	**8**	**3**	**78**

CREDIT DISTRIBUTION: SEMESTER WISE

Sr. No.	Type of course	No. of Credits/Semester								Total
		1	2	3	4	5	6	7	8	
1	Basic Science Course	9	9	5						**23**
2	Engineering Core/Science Course	7	9	3	3					**22**
3	Humanities, Social Sciences and Management Course	2	2	3	2	2	2			**13**
4	Professional Core Course			12	12	8	8	8		**48**
5	Professional Elective Course					6	6	6		**18**
6	Open Elective Course				3	3	6	6		**18**
7	Project	2							14	**16**
8	Internship								3	**3**
9	Audit Course									**0**
10	Mandatory Course									**0**
11	Life Skills									**0**
12	Proficiency Course									**0**
13	MOOC									**0**
Total		**20**	**20**	**23**	**20**	**19**	**22**	**20**	**17**	**161**

B. Tech (Information Technology) Curriculum Structure Semester III

		Semester – III	Teaching Scheme					Evaluation Scheme						
Course Code	Category	Course Name	L	T	P	H	Cr	IE	MTE	ETE	TW	PR	OR	Total
BAS3205	BSC	Applied Mathematics	3	-	-	3	3	20	30	50	-	-	-	100
BAS3206	BSC	Data Analysis using R	-	-	4	4	2	-	-	-	50	-	-	50
BIT3301	ECC	Logic Design and Computer Organization	2	-	-	2	2	20	30	50	-	-	-	100
BIT3401	PCC	Discrete Mathematics	2	1	-	3	3	20	30	50	-	-	-	100
BIT3402	PCC	Fundamentals of Data Structures	3	-	-	3	3	20	30	50	-	-	-	100
BIT3403	PCC	Data Communication and Computer Networks	3	-	-	3	3	20	30	50	-	-	-	100
BIT3302	ECC	Logic Design and Computer Organization-Lab	-	-	2	2	1	-	-	-	25	-	-	25
BIT3404	PCC	Fundamentals of Data Structures – Lab	-	-	4	4	2	-	-	-	50	25	-	75
BIT3405	PCC	Data Communication and Computer Networks Lab	-	-	2	2	1	-	-	-	25	-	25	50
BHM3101	HSMC	Universal Human Values	3	-	-	3	3	30	-	20	-	-	-	50
BHM3939	LS	Life Skill-III	0	0	2	2	-	-	-	-	-	-	-	0
Total			**16**	**01**	**14**	**31**	**23**	**130**	**150**	**270**	**150**	**25**	**25**	**750**

B. Tech (Information Technology) Curriculum Structure Semester IV

		Semester – IV	Teaching Scheme					Evaluation Scheme						
Course Code	Category	Course Name	L	T	P	H	Cr	IE	MTE	ETE	TW	PR	OR	Total
BIT4301	ECC	Microprocessors and Microcontrollers	2	-	-	2	2	20	30	50	-	-	-	100
BIT4401	PCC	Data Structures & Algorithms	3	-	-	3	3	20	30	50	-	-	-	100
BIT4402	PCC	Object Oriented Programming with JAVA	2	-	-	2	2	20	30	50	-	-	-	100
BIT4403	PCC	Automata Theory	3	1	-	4	4	20	30	50	-	-	-	100
BIT4302	ECC	Microprocessors and Microcontrollers –Lab	-	-	2	2	1	-	-	-	25	25	-	50
BIT4404	PCC	Data Structures & Algorithms-Lab	-	-	2	2	1	-	-	-	25	25	-	50
BIT4405	PCC	Object Oriented Programming-Lab	-	-	4	4	2	-	-	-	50	25	-	75
BAS4601 to BAS4606	OEC	Open Elective-I	3	-	-	3	3	20	30	50	-	-	-	100
BHM4101	HSMC	Professional Skills for Engineers	2	-	-	2	2	30	-	20	-	-	-	50
BIT4911	PFC	Proficiency Course-I	-	-	2	2	-	-	-	-	-	-	-	0
BHM4940	LS	Life Skill-IV	-	2	-	2	-	-	-	-	-	-	-	0
BHM9961	AC	Audit Course	1	-	-	1	-	-	-	-	-	-	-	0
Total			**16**	**03**	**10**	**29**	**20**	**130**	**150**	**270**	**100**	**75**	**0**	**725**

Abbreviations: Course Abbreviation; **L**- Lecture; **P**- Practical; **H**- Hours; **CR**- Credits; **IE**– Internal Evaluation-1; **MTE** – Mid Term Examination **ETE** – End Term Examination; **TW** – Term Work; **OR** – Oral Exam.

List of Open Electives
Semester- IV

Course Code	Course Name	
BAS4601	Numerical Methods	Choose any one
BAS4602	Mathematical Optimization	
BAS4603	Calculus of Variation	
BAS4604	Mathematical Modelling and Simulation	
BAS4605	Financial Mathematics	
BAS4606	Neural Network and fuzzy logic Control	

List of Proficiency Courses
Semester- IV

Course Code	Course Name	
BIT4911A	Python Programming	Choose any one
BIT4911B	Web Development	

List of Life Skill Courses
Semester- III

Course Code	Course Name: Life Skills-III	
BHM3939	1. Practicing Meditation 2. Sports	Choose any one
	Performing Arts: Music, Singing, Poetry, Indian Conventional Dancing, Photography, Short Movie Making, Painting/ Sketching/ Drawing, Theatre Arts, Anchoring, Calligraphy etc.	Choose any one performing arts

Semester- IV

Course Code	Course Name: Life Skills-IV	
BHM4940	1. Social welfare and Cultural Awareness 2. Transactional Analysis	Choose any one
	Caring and service Hospital Caring, Personal Safety, First Aid, Disaster Management Gardening, Organic farming, Cooking etc.	Choose any one caring & service

List of Audit Courses
Semester –III

Course Code	Name of Course
BHM9961	Environmental Science

Course Syllabus

Semester-III

Program:	**B. Tech. I.T.**				Semester: **III**		
Course: **Logic Design and Computer Organization**					Code: **BIT3301**		
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
2	-	2	2	20	30	50	100

Prior Knowledge of :

1. Analog and Digital Electronics
2. Logic gates

is essential

Course Objectives:

1. To make undergraduates, aware of different levels of abstraction of computer systems from hardware perspective.
2. To learn and understand basic digital design techniques.
3. To develop design and implementation skills of combinational and sequential logic circuits.
4. To make undergraduates understand the functions, characteristics of various components of Computer: processor & memory.

Course Outcomes:

After learning the course, the students will be able to:

1. Recall the knowledge of number systems, Boolean algebra, logic gates and code conversions, use of logic function representation for simplification with K-Maps.
2. Demonstrate Combinational logic circuit design using SSI & MSI chips.
3. Develop Sequential circuits like Flip-Flops (Truth Table, Excitation table), their conversion & applications.
4. Summarize the functions & organization of various blocks of Computer Architecture, different types of Computer Architectures.

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1.	**Number System and Logic Minimization** Number Systems: Binary, Octal, Hexadecimal and their conversions. signed binary number representation, complement representation and arithmetic's. Codes: BCD, Excess-3, Gray code, Binary Code and their conversions. Logic minimization: Representation of truth-table, SOP form, POS form, reduction techniques: K-Maps up to 4 variables, don't care conditions.	6
2.	**Combinational Logic Design** Design using SSI chips: Code Converters, Adders, Subtractors, n bit Binary Adder. Introduction to MSI chips: Multiplexer (IC 74153), De-multiplexer (IC 74138), Decoder (74238) Encoder (IC 74147), Binary adder (IC 7483).	6
3.	**Sequential Logic Design** Introduction to sequential circuits: Difference between combinational and sequential circuits; Memory element-Latch & Flip-Flop. Flip- Flops: Logic diagram, truth table & excitation table of SR, JK, D, T flip flops; Conversion from one FF to another, Study of flip flops with regard to asynchronous and	6

	synchronous, Preset & Clear. Application of flip-flops**:** Counters- Asynchronous, Synchronous and Modulo N Counters, Registers- shift register types& applications.	
4.	**Computer Organization** Computer organization & computer architecture: organization, functions & types of computer units- CPU (typical organization, Functions, Types), Memory (Types & their uses in computer), IO (types & functions) & system bus. (Address, data & control bus), ALU (ALU signals, functions & types); Register (types & functions of user visible, control & status registers such as general purpose, address registers, data registers, flags, PC, MAR, MBR, IR) & control unit. Types of Computer Architectures: Von Neumann & Harvard architecture, RISC and CISC Architectures.	6
	Total	**24**

Text Books:

1. R.P. Jain, "Modern Digital Electronics", 3rd Edition, Tata McGraw-Hill, ISBN: 0–07–049492–4.
2. William Stallings, "Computer organization and architecture, designing for performance" Prentice Hall, Eighth edition.

Reference Books:

1. Floyd, "Digital Principles", Pearson EducationISBN:978-81- 7758-643-6.
2. M Morris Mano, "Digital Design", Prentice Hall, 3rd Edition, ISBN: 0130621218.
3. John Yarbrough, "Digital Logic applications and Design", Thomson Publication ISBN: 978-0314066756.
4. Malvino, D. Leach, "Digital Principles and Applications", 5th edition, Tata McGraw Hill.
5. Hamacher and Zaky, "Computer organization", Fifth Edition.
6. D. Patterson, J. Hennessy, "Computer Organization and Design: The Hardware Software Interface" Fourth Edition, Morgan Kaufmann.

Program:	**B. Tech. I.T.**				Semester: **III**		
Course: **Discrete Mathematics**					Code: **BIT3401**		
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
2	1	3	3	20	30	50	100

Prior Knowledge of:

1. Basic Mathematics concepts**.**

is essential

Course Objectives:

1. To gain sound knowledge to formulate and solve problems with sets and propositions.
2. To understand and solve counting problems by applying elementary counting techniques to solve problems of discrete probability.
3. To understand Graph and Tree terminologies and models to be applied in real life problems.
4. To recognize types of relation, formulate and solve problems with relations and functions.
5. To understand basics of number theory and its applications.
6. To understand the various types' algebraic structures and its applications.

Course Outcomes:

After learning the course, the students will be able to:

1. Formulate and apply formal proof techniques and solve the problems with logical reasoning.
2. Analyse and evaluate the combinatorial problems by using probability theory.
3. Apply the concepts of graph theory to devise mathematical models.
4. Analyse types of relations and functions to provide solution to computational problems.
5. Identify techniques of number theory and its application.
6. Identify fundamental algebraic structures.

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1.	**Sets And Propositions** Sets: Sets, Combinations of Sets, Venn Diagram, Finite and Infinite Sets, Countable Sets, Multisets, Principle of Inclusion and Exclusion. Proof Techniques: Direct and Indirect Methods, Mathematical Induction, Propositions: Propositions, Logical Connectives, Conditional and Bi-conditional.	8
2.	**Combinatorics And Discrete Probability** Combinatorics: Rules of Sum and Product, Permutations, Combinations. Discrete Probability: Discrete Probability, Conditional Probability, Bayes Theorem, Applications of Combinatorics and Discrete Probability.	6
3.	**Relations & Functions** Relations**:** Properties of Binary Relations, Closure of Relations, Warshall's Algorithm, Equivalence Relations, Partitions, Partial Ordering Relations, Lattices. Functions: Functions, Composition of Functions, Invertible Functions, Pigeonhole Principle, Discrete Numeric Functions.	6
4.	**Graph Theory** Graphs: Basic Terminologies, Multi-Graphs, Weighted Graphs, Sub Graphs, Isomorphic graphs, Complete Graphs, Regular Graphs, Bipartite Graphs, Operations on Graphs, Paths, Circuits, Hamiltonian and Eulerian graphs, Travelling Salesman Problem, Factors of Graphs, Planar Graphs, Graph Colouring.	5

5.	**Trees** Trees**:** Tree Terminologies, Rooted Trees, Path Length in Rooted Trees, Prefix Codes, Spanning Trees, Fundamental Cut Sets and Circuits, Max flow –Min Cut Theorem (Transport Network). Binary Trees and traversal, Applications of Graph Theory.	5
6.	**Elementary Number Theory** Divisibility of Integers: Properties of Divisibility, Division Algorithm, Greatest Common Divisor GCD and its Properties, Euclidean Algorithm, Prime Factorization Theorem, Congruence Relation, Modular Arithmetic, Euler Phi Function, Euler's Theorem, Fermat's Little Theorem, Additive and Multiplicative Inverses, Chinese Remainder Theorem. Algebraic Structures: Introduction Semigroup, Monoid, Group, Abelian Group, Permutation Groups, Ring, Integral Domain, Field. Applications of Algebraic Structures.	6
	Total	**36**

Text Books:

1. Kenneth H. Rosen. Discrete Mathematics and Its Applications, 7th Edition, McGraw Hill, 2012.
2. C. L. Liu and D. P. Mohapatra, "Elements of Discrete Mathematics", 4th Edition, McGraw-Hill.

Reference Books:

1. Bernard Kolman, Robert C. Busby, Sharon Cutler Ross, "Discrete mathematical structures", 6th edition, Prentice Hall of India.
2. Edgar G. Goodaire, Michael M. Parmenter, "Discrete Mathematics with Graph Theory", 3rd Edition, Pearson Education.
3. Tremblay J. S., "Discrete mathematical structures with application", 3rdEdition, Tata McGraw Hill.
4. Lipschutz Seymour, "Discrete mathematics", 4th Edition, Tata McGraw-Hill.
5. Johnsonbaugh Richard, "Discrete Mathematics", 7th edition, Pearson.
6. Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science", Dover publication.
7. Introduction to Discrete Mathematics for Computer Science Specialization by Coursera, https://www.coursera.org/specializations/discrete-mathematics.
8. "Discrete Mathematics", Master Discrete Math for Computer Science and Mathematics Students by Udemy.

Program:	**B. Tech. I.T.**					Semester**: III**	
Course: **Fundamentals of Data Structures**						**Code: BIT3402**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
3	0	3	3	20	30	50	100

Prior knowledge of:

1. Programming and problem-solving skills

is essential

Course Objectives:

1. To apply C language constructs and pointers in depth.
2. To analyze algorithms with respect to time and space complexity.
3. To select appropriate searching and/or sorting techniques in the application development.
4. To learn and apply linear data structures and their applications.

Course Outcomes:

After learning the course, the students will be able to:

1. Apply pointers, dynamic memory allocation concepts, and file handling in application development.
2. Analyze algorithms with respect to time and space complexity.
3. Compare and select appropriate searching and/or sorting techniques in the application development.
4. Construct Linear data structures using arrays and linked list
5. Make use of abstract properties of stack data structures
6. Construct abstract properties of various queue data structures.

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1.	**Pointers in C and File Handling** Introduction to Pointers, pointer to pointer, pointer to single and multidimensional arrays, array of pointers, pointer to functions, dynamic memory allocation, Pointer to file structure and basic operations on file, single and multiple file handling in C.	5
2.	**Analysis of Algorithms and Searching and Sorting Techniques** Analysis of algorithm: frequency count and its importance in the analysis of an algorithm, Time complexity & Space complexity of an algorithm, Big 'O', 'Ω' and 'Θ' notations, Best, Worst and Average case. Need of searching and sorting, complexity analysis of linear search, binary search, Concept of internal and external sorting, sort stability, bubble sort, insertion sort, selection sort, Merge sort and quick sort, Complexity analysis of sorting techniques.	7
3.	**Linear Data Structures Using Sequential Organization** Concept of data structures, Concept of sequential organization, Multidimensional arrays and their storage representation: row-major and column-major form and address calculation. Representation of sparse matrix using arrays, algorithms for sparse matrix simple and fast transpose.	5

4.	**Linear Data Structures Using Linked Organization** Concept of linked organization, singly linked list, doubly linked list, circular linked list, Generalized Linked List, Representation and manipulations of polynomials using linked lists.	7
5.	**Stacks** Concept of the stack, stack as an ADT, Implementation of stack using linked organization. Concept of the implicit and explicit stack, Applications of the stack: reversing the list, expression conversion, and evaluation.	7
6.	**Queues** Concept of queues as ADT, Implementation of queue using linked organization. Concept of the circular queue, double-ended queue, and priority queue. Applications of queues.	5
	Total	**36**

Text Books:

1. E. Horowitz, S. Sahani, S. Anderson-Freed "Fundamentals of Data Structures in C", Universities Press, (2008).
2. Richard F. Gilberg, Behrouz A. Forouzan, “Data Structures: A Pseudocode Approach with C”, Second Edition, Cengage Learning, ISBN 9780534390808 (2004).

Reference Books:

1. Yashwant Kanetkar, “Understanding Pointers in C”, 4th Publication, BPB Publication, ISBN: 9788176563581 (2003).
2. G. A.V. PAI, “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6.
3. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
4. A. Tharp,” File Organization and Processing”, Willey India edition, 9788126518685, (2008).
5. J. Tremblay, P. Soresan, "An Introduction to Data Structures with Applications", 2nd edition, Tata McGraw Hill International Editions, 1984, ISBN-0-07-462471-7.

Program:	**B. Tech. I.T.**					Semester: **III**	
Course: **Data Communication and Computer Networks**						Code: **BIT3403**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
3	-	3	3	20	30	50	100

Prior Knowledge of:

1. Basic Mathematics
2. Fundamentals of communication

is essential

Course Objectives:

1. To gain the knowledge of communication systems.
2. To Learn and understand the History of Computer Network and its evolution with the help of service models
3. To learn the various issues of Network layer, its management and Routing algorithms at Network layer
4. To learn the services offered by Transport Layer
5. To learn the session layer and Applications Layer protocols; and its services.
6. To Learn and understand the Internet and Web functionality with its protocols

Course Outcomes:

After learning the course, the students will be able to:

1. Identify various Data Communication Techniques and its trade off.
2. Specify and identify existing topological structures and protocols.
3. Analyse routing strategies for an IP based networking infrastructure
4. Evaluate the challenges in building networks and solutions to those.
5. Gain the knowledge of datagram and Internet.
6. Use and apply the standards and protocols learned, for application development

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1.	**Introduction to Data Transmission and Modulations** Data Transmission methods – Synchronous & Asynchronous, Flow Control, Error Control, Error Detection and correction, Overview of Data & Signal Bits. Baud & Bit Rate. Analog Modulations Techniques, Digital Modulation Techniques. Types of Noise: External Noise, Internal Noise. Communication Channel - Discrete and Continuous Channel, Shannon-Hartley Theorem, Channel Capacity, Nyquist and Shanon Theorem, Bandwidth S/N Trade Off.	8
2.	**Introduction to Computer Networks** Introduction to OSI and TCP/IP Protocol Suite, Classification of Addressing Mechanisms, Types of Networks-LAN, WAN And MAN, Network Topologies. Guided Media: Twisted-Pair Cable, Coaxial Cable and Fiber-Optic Cable, Unguided Media: Wireless, Radio Waves, Microwaves and Infrared, Wireless frequency spectrum (EU and American Standard).	5
3.	**Network Layer and Routing Principles** Design issues, virtual circuit and datagram subnet, Study of Router, Routing algorithms like adaptive and non-adaptive, Congestion control algorithms and prevention policies, Load shading, Traffic Management, jitter control, quality of service, internetworking, Routing Algorithms.	5

4.	**Transport Layer and its Services** Overview of Transport layer, Transport layer service primitives, Multiplexing and De-multiplexing, Connection oriented (TCP) and Connection less transport (UDP) protocol, Flow control-Retransmission, TCP Congestion control, Congestion avoidance, Connection management, QoS and Application requirements.	6
5.	**Session and Application Layer** Overview of Session Layer and Services Offered, Role of Session Layer in OSI Model, Session Control Protocols, Session Management in TCP. Introduction to Application Layer, OSI vs TCP/IP, WWW & HTTP, FTP, TFTP, SMTP, POP3, IMAP, MIME, P2P.	6
6.	**Internet and Web** Internet Addressing Scheme, IPv4 vs IPv6, Network Byte Order, Domain name system (DNS) and DNS servers, Resource records, Introduction to Internet security and Firewall, Website Planning and Hosting.	6
	Total	**36**

Text Books:

1. Behrouz A Forouzan , “Data Communications and Networking”, 4th Ed, Tata McGraw – Hill, 2011.
2. S. Tanenbaum : "Computer, Networks", PHI Publication, 4th edition, ISBN: 8178087855

Reference Books:

1. H. Taub And K.L. Shiling, “Principles of Communication System”, 3rd Edition, Tata Mcgraw Hill Education Private Limited.
2. James F. Kurose, Keith W. Ross, “Computer Networking – A Top-Down Approach Featuring the Internet”, Fifth Edition, Pearson Education, 2009.
3. Behrouz A. Forouzan,”TCP-IP protocol suite “, Tata McGraw Hill Edition, 2nd edition,2003, ISBN: 978007060004.

Program:	**B. Tech. I.T.**				Semester: **III**		
Course: **Logic Design and Computer Organization Laboratory**					Code: **BIT3302**		
Teaching Scheme				**Evaluation Scheme**			
Practical	**Tutorial**	**Credit**	**Hours**	**TW**	**OR**	**PR**	**Total**
2	-	1	2	25	-	-	25

Course Objectives:

1. To design & implement combinational and sequential circuits.
2. To learn simulation of digital systems.
3. To Study Basic Computer Architecture.

Course Outcomes:

After learning the course, the students will be able to:

1. Illustrate the concepts of logic function representation for simplification with K-Maps.
2. Construct Combinational & Sequential logic circuits.
3. Develop circuits of Modulus and Ripple counters using synchronous counters.
4. Demonstrate simulation of basic Computer blocks such as ALU & memory.

Detailed Syllabus	
Assignment No.	**Suggested List of Assignments**
GROUP A **Combinational Logic Design (Any 4)**	
1.	Realize Full Adder and Subtractor using a) Basic Gates and b) Universal Gates.
2.	Design and implement 4-bit BCD to Excess-3 code.
3.	Design and implement 1 digit BCD adder using IC7483.
4.	Design and implement following using multiplexer IC 74153 1) full adder 2) Any three-variable function (cascade method).
5.	Design and implement full subtractor using decoder IC 74138.
GROUP B **Sequential Logic Design**	
6.	Design and implement 3 bits Up and 3 bit Down Asynchronous Counters using master slave JK flip-flop IC 7476.
7.	Design and implement 3 bits Up and 3 bit Down Synchronous Counters using master slave JK flip-flop IC 7476.
8.	Design and implement Modulo ‘N’ counter using IC7490. (N= 100 max).
9.	Design of Ripple Counter using suitable Flip Flops.
GROUP C	

Computer organization (Any 1 using virtual lab simulator)	
10.	Design & simulate single bit ALU with four functions (AND, OR, XOR, ADD).
11.	Design & simulate single bit RAM cell.
GROUP D **Study Practical (Any 1)**	
12.	Study of Shift Registers (SISO,SIPO, PISO,PIPO).
13.	Study of Basic Computer Architecture in detail.
Remark: Implementation of all assignments is encouraged.	

Reference Books:

1. R.P. Jain, "Modern Digital Electronics", 3rd Edition, Tata McGraw-Hill, ISBN:0-07-049492-4
2. William Stallings, “Computer organization and architecture, designing for performance” ,Prentice Hall, Eighth edition
3. Virtual Lab simulator Link http://vlabs.iitkgp.ac.in/coa/

Program:	**B. Tech. I.T.**					Semester: **III**	
Course: **Fundamentals of Data Structures Laboratory**						Code: **BIT3404**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credit**	**TW**	**OR**	**PR**	**Total**
-	4	-	2	50	-	25	75

Course Objectives:

1. To write a neat code by following coding standards by selecting appropriate data structures.
2. To analyze algorithms with respect to time and space complexity.
3. To select appropriate searching and/or sorting techniques in the application development.
4. To implement abstract properties of various data structures such as stacks, queues.
5. To identify suitable data structures and algorithms to solve real-world problems.

Course Outcomes:

After learning the course, the students will be able to:

1. Make use of coding standard for application development.
2. Perform analysis of algorithms with respect to time and space complexity.
3. Select appropriate searching and/or sorting techniques in the application development.
4. Construct Linear data structures using arrays and linkedlists.
5. Make use of abstract properties of various data structures such as stacks, queues.
6. Identify suitable data structure and algorithm to solve a real-world problem.

Assignment No.	**Suggested List of Assignments** (Write a program in C)
	GROUP A
1.	**Assignments for the practice of language syntax, data types, and programming constructs** **(Any 4, Practice Program)** a) To simulate a simple calculator that performs basic tasks such as addition, subtraction, multiplication, and division. b) Pass two parameters to function as(start and end), and display all prime numbers between two Intervals. c) Create class STORE to keep track of Products (Product Code, Name, and price). Display menu of all products to the user. Generate bill as per order. d) Accept a string and change the case of each character of the string. Example “THIs Is a C Program” changes to “thiSiS A c pROGRAM”. e) Accept a string from the user and perform the following string operations – i. Calculate the length of string ii. String reversal iii. Equality check of two strings iv. Check palindrome v. Check substring (Pointer to strings). Represent matrix using two-dimensional arrays and perform the following operations without using pointers: i. Addition ii. Multiplication.

GROUP B (Any 8)	
2.	Represent sets using one-dimensional arrays and implement functions to perform i. Union ii. Intersection iii. Difference iv. Symmetric difference of two sets
3.	Represent matrix using two-dimensional arrays and perform the following operations with pointers: i. Addition ii. multiplication iii. transpose iv. Saddle point
4.	Write a Menu driven program in C for a cricket player's display board. The information of the cricketer can be (not limited to) Name, Age, Country, Category (Batsman, Bowler, Wicket keeper, All-rounder), Number of ODI's played, Number of International 20-20's played. Display following. a) Number of batsman of a particular country b) Sort the Batsman as per the average batting score(Bubble/insertion sort) c) Batsman with highest average score (Binary search) d) Number of bowlers of a particular country e) The bowler that has taken a maximum no of wickets f) Show particular players the entire "Display board information" Delete/Modify the record
5.	Implement Quick Sort to sort the given list of numbers. Display corresponding list in each pass.
6.	Implement a sparse matrix with operations like initialize empty sparse matrix, insert an element, add two matrices, transpose a matrix.
7.	Implement a Polynomial with operations like create an empty polynomial, insert an entry into a polynomial, add two polynomials and evaluate a polynomial.
8.	Department of Information Technology has a student's club named 'Pinnacle Club'. Students of the Second, third and final year of the department can be granted membership on request. Similarly, one may cancel the membership of the club. The first node is reserved for the president of the club and the last node is reserved for the secretary of the club. Write C program to maintain club member's information using a singly linked list. Store student PRN and Name. Write functions to a) Add and delete the members as well as the president or even secretary. b) Compute the total number of members of the club c) Display members of the club Display list in reverse order using recursion
9.	Implement any database using a doubly-linked list with the following options a) Insert a record b) delete a record c) modify a record d) Display list forward Display list backward
10.	To convert decimal numbers to binary numbers using the stack as sequential or linked representation.
11.	Implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix, and evaluation of postfix and prefix expression.
12.	Implement Circular Queue using Array. Perform following operations on it. a) Insertion (Enqueue) b) Deletion (Dequeue) Display

13.	Implement priority queue as ADT using a single linked list for servicing patients in a hospital with priorities as a) Serious (top priority) b) medium illness (medium priority) General (Least priority)
14.	Department maintains student information. The file contains roll number, name, division, and address. Write a program to create a sequential file to store and maintain student data. It should allow the user to add, delete information of the student. Display information of the particular employee. If the record of the student does not exist an appropriate message is displayed. If a student record is found it should display the student details.
GROUP C (Mini-project)	
15.	Mini-project: Write an application demonstrating your skills in defining a problem, writing down the requirements carefully, designing a modular solution with clear separation of abstract data types and their use, design of proper function prototypes, and division of work among functions. The application can be defined after discussion with the course instructor.

Remark: Assignment 1 (week 1) is for practice of all prerequisites. Hence, this assignment is not included in the Practical Exam.

Reference Books:

1. Richard Reese - Understanding and Using C Pointers-O'Reilly Media (2013).
2. G. A.V. PAI, “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6.
3. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
4. A. Tharp,” File Organization and Processing”, Willey India edition, 9788126518685, (2008).
5. J. Tremblay, P. Soresan, "An Introduction to Data Structures with Applications", 2nd edition, Tata McGraw Hill International Editions, 1984, ISBN-0-07-462471-7.
6. Amiya Kumar Rath, Alok Kumar Jagdev, “Data Structures using C”, Scitech Publication, 2nd Edition, ISBN: 978-81-8371-2323.
7. ds1-iiith.vlabs.ac.in/exp/stacks-queues/index.html
8. ds1-iiith.vlabs.ac.in/exp/poly-arithmetic/index.html

Program:	**B. Tech. I.T.**					Semester: **III**	
Course: **Data Communication and Computer Networks Laboratory**						Code: **BIT3405**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Credit**	**Hours**	**TW**	**OR**	**PR**	**Total**
-	2	1	2	25	25	-	50

Course Objectives:

1. To study installation, maintenance of network Laboratory based on Computer Networks Technology
2. To design network by using various topologies
3. To Learn and apply the TCP/IP commands
4. To learn and use Network Simulators
5. To learn various Server application

Course Outcomes:

After learning the course, the students will be able to:

1. Design and install various Network topologies
2. Apply various Commands to gain the information of the installed Network and Internet
3. Design the networks using the simulators
4. Install and configure various server services.
5. Explore use of protocols in various web applications.
6. Develop applications on emerging trends.

Detailed Syllabus	
Unit	**Description**
1.	Network Design by using various Topologies in CISCO Packet tracer
2.	Basic TCP/IP utilities and commands. (eg: ping, ifconfig, tracert, arp, tcpdump, whois, host, netsat, nslookup, ftp, telnet etc…)
3.	Design of • Small Network • Large Network using CISCO Packet Tracer / NS3.0 Simulators
4.	Installation and Configuration of the Following • Remote Login Service – TELNET/SSH • Configuration of FTP server and accessing it via FTP Client. • Installation and configuration of APACHE WEB SERVER / IIS • PWS along with HTTP server. • Installation and configuration of DHCP Server in Wireless Environment using an Access Point. • Installation and configuration of DNS Server. Installation and configuration of Mail Server.

Instruments Required:

1. LAN Trainer Kit
2. Unmanaged Switch
3. Router
4. Access Point

Reference Books:

1. James F. Kurose, Keith W. Ross, “Computer Networking – A Top-Down Approach Featuring the Internet”, Fifth Edition, Pearson Education, 2009.
2. Behrouz A. Forouzan, “Data communication and Networking”, Fourth Edition, Tata McGraw – Hill, 2011.
3. S. Tanenbaum : "Computer, Networks", PHI Publication, 4th edition, ISBN: 8178087855.
4. Behrouz A. Forouzan,”TCP-IP protocol suite “, Tata McGraw Hill Edition, 2nd edition,2003, ISBN: 978007060004.

List of Open Source Software/learning website:

Wireshark packet analyzer,

1. NS3.0/CISCO network simulator
2. Netsim

Course Syllabus

Semester-IV

Program:	B. Tech. (I.T.)					Semester : IV	
Course : Microprocessors and Microcontrollers						Code :**BIT4301**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
2	-	2	2	20	30	50	100

Prior Knowledge of :

1. Number system and logic minimization
2. Computer organization & computer architecture

is essential

Course Objectives:

1. To study architecture and features of 8086 microprocessors.
2. To know the programming with 8086 concepts.
3. To study architectural details of PIC18 microcontroller.
4. To introduce the concept of PIC I/O Ports and Timer.

Course Outcomes:

After learning the course, the students should be able to:

1. Explain basics of assembly language programming & 8086 architecture.
2. Illustrate programming concepts of 8086 like register set, instruction set & interrupt handling.
3. Interpret architecture and memory organization of PIC18 microcontroller.
4. Utilize concepts of I/O ports and timers of PIC18.

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1.	**Introduction To Assembly Language Programming & 8086 Processor** Introduction to assembly language programming, ALP tools- Assembler, Linker, Loader, Debugger, Emulator, Assembler directives, Far and near procedure, Macros, DOS Internals, DOS Calls. Introduction to microprocessor concept, 8086 - Features and Architecture.	**6**
2.	**Programming With 8086** Pin out Diagram, Register Set, 8086 Real mode segmentation and Address translation, Addressing modes, Instruction set.8086 Interrupt structure, ISR.	**6**
3.	**PIC Microcontroller Architecture** Introduction: Microcontroller, Difference between microprocessor and microcontroller, Criteria for selection of microcontroller, PIC18F: Features and architecture, Pin out Diagram, Registers set. Program and data memory organization: The Program Counter and Programmable ROM space in the PIC, File register and Access bank, Bank switching; Addressing modes: Addressing modes with instruction example, Reset operations, Brownout reset, Watchdog timer & Configuration registers.	**6**
4.	**PIC I/O Ports and Timer** I/O Port: I/O Port structure with programming: I/O Port structure, I/O Port programming, I/O Bit manipulation Programming. Timer/Counter: Registers used for Timer/Counter operation, Delay calculations, Programming of Timers using Embedded C. Case Study Traffic light signal controller using Timer/Counter.	**6**
	Total	**24**

Text Books:

1. James Turley, "Advanced 80386 Programming Techniques", McGraw Hill Education.
2. Muhammad Ali Mazidi , Danny Causey, RolinMcKinlay, “PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18”, 4th Edition by,Pearson international edition.

Reference Books:

1. Walter A. Tribel, Avtar Singh, "The 8088 and 8086 Microprocessors", 4th edition, Prentice Hall of India.
2. Ray Duncan, "Advanced MS DOS Programming", 2nd edition, BPB Publications.
3. Peatman, John B, “Design with PIC Microcontroller”, Pearson Education PTE.
4. Ramesh Gaonkar, “Fundamentals of Microcontrollers and Applications In Embedded Systems(with the PIC18 Microcontroller Family)”Thomson/Delmar Learning; 1 edition (January 8, 2007)ISBN:978-1401879143.

Program:	**B. Tech. I.T.**			Semester: **IV**			
Course: **Data Structures and Algorithms**				Code: **BIT4401**			
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
3	-	3	3	20	30	50	100

Prior knowledge of:

1. Pointers
2. Complexity analysis
3. Linear data structures
4. Basics of trees and graphs

is essential

Course Objectives:

1. To apply various advanced data structures such as trees and graphs to the applications.
2. To make use of Symbol tables and hashing for efficient searching.
3. To know the basics of computational complexity analysis.
4. To describe, apply and analyze the complexity of algorithmic design strategies.
5. To provide a thorough knowledge of the most common algorithms and data structures.

Course Outcomes:

After learning the course, the students will be able to:

1. Apply algorithms to create and traverse non-linear data structure trees.
2. Apply graphs to various computing problems.
3. Make use of Symbol tables and hashing for efficient searching.
4. Apply Divide & Conquer as well as Greedy approach to design algorithms.
5. Illustrate different problems using Backtracking and Branch and Bound strategy.
6. Explore the concept of P, NP, NP-complete, NP-Hard and parallel algorithms.

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1.	**Trees** The difference in the linear and non-linear data structure, Trees, and binary trees-concept and terminology. Expression tree. Conversion of the general tree to binary tree. Binary tree as an ADT. Recursive and non-recursive algorithms for binary tree traversals, Binary search trees, Binary search tree as ADT, Applications of trees.	5
2.	**Graphs** Graph as an ADT, Representation of graphs using adjacency matrix and adjacency list, Depth First Search and Breadth-First Search traversal. Prim's and Kruskal's algorithms for minimum spanning tree, shortest path algorithm, topological sorting.	5
3.	**Symbol Tables & Hashing:** Symbol Table notion, Concept of red and black trees, AVL Trees, Heap data structure, Min and Max Heap, Heap sort implementation, Hashing, hashing functions, collision resolution techniques.	6

4.	**Divide And Conquer** Solving Recurrence Equations - Homogeneous and non-homogeneous. Divide & Conquer: General method, Control abstraction, Finding Max-Min, Large integer Multiplication (for all above algorithms analysis to be done with recurrence).	6
5.	**Greedy Method and Dynamic Programming:** Greedy Method: General method and characteristics, Optimal storage on tapes, Fractional Knapsack problem. Dynamic Programming:General strategy, Principle of optimality, Multistage Graph problem, Travelling Salesman Problem.	6
6.	**Backtracking, Branch-and-bound and Complexity Theory:** Backtracking:General method, Recursive backtracking algorithm, Iterative backtracking method. 8-Queen problem, Sum of subsets, Graph coloring. Branch-and-bound:LC branch-and-bound, FIFO branch-and-bound. Complexity Theory:The classes P, NP, NP complete, NP hard satisfiability problem, proofs for NP complete problem, Clique, Vertex cover problem.	8
	Total	**36**

Text Books:

1. E. Horowitz, S. Sahani, S. Anderson-Freed "Fundamentals of Data Structures in C", Universities Press, 2008.
2. Ellis Horwitz, Sartaj Sahni,Raj Sekaran. Fundamentals of computer algorithms University Press.

Reference Books:

1. Thomas H Cormen and Charles E.L Leiserson, Introduction to Algorithm, PHI, ISBN:81-203-2141-3.
2. G. A.V, PAI, “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6.
3. Gilles Brassard, Paul Bratle, Fundamentals of Algorithms, Pearson, ISBN 978-81-317-1244.
4. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.

Program:	**B. Tech. I.T.**						Semester: **IV**
Course: **Object Oriented Programming with JAVA**							Code: **BIT4402**
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
2	-	2	2	20	30	50	100

Prior knowledge of:

1. Problem solving skills
2. Procedure Oriented Programming

is essential

Course Objectives:

1. To teach fundamentals of object-oriented concepts and programming
2. To apply the concepts of object-oriented paradigm.
3. To develop object oriented programming skills
4. To design and implement applications for real life problems by using object oriented programming.

Course Outcomes:

After learning the course, the students will be able to:

1. Explain features of JAVA and object oriented programming
2. Construct classes for given real life data and operations.
3. Design hierarchy of classes using inheritance.
4. Develop application which handles different types of exceptions.

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1	**Foundations of Object Oriented Programming** Object Oriented Paradigm, Features of Object-Oriented Programming: Objects and Classes, Data Abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Communication. Features of Java, Overview of Java.	6
2	**Classes, Methods and Constructors** Class Fundamentals, Methods, Constructors, Instance variable hiding, Garbage Collection, Finalize Method, Method Overloading, Object as a Parameter, Returning Object, Visibility/Access Control, Static Variable and Methods.	6
3	**Inheritance and Polymorphism** Inheritance Basics, Use of Super, Constructors in derived Classes, Method Overriding, Dynamic Method Dispatch, Abstract Classes, and Using Final with inheritance, Interfaces and Packages.	6
4	**Exception Handling** Fundamentals, Exception Types, Using try and catch, throw, throws and finally, multiple catch clauses, Nested try statements, built in exceptions.	6
	Total	**24**

Text Books:

1. E. Balagurusamy, “Programming with Java – A Primer”, Tata – McGraw-Hill Publication, 4th Edition, 2019.
2. Steven Holzner et al. “Java 2 Programming”, Black Book, Dreamtech Press, 2009.

Reference Books:

1. Java The complete reference, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd. 9th edition, 2014, ISBN: 978-0-07-180856-9 (E-book).
2. MitsunoriOgihara, “Fundamentals of Java Programming”, Springer; 2018, ISBN 978-3-319-89490-4.
3. Brahma DathanSarnath Ramnath, “Object-Oriented Analysis, Design and Implementation An Integrated Approach”, Springer; 2nd ed. 2015, ISSN 1863-7310 ISSN 2197-1781 (electronic) Undergraduate Topics in Computer Science ISBN 978-3-319-24278-1, ISBN 978-3-319-24280-4.
4. Object-Oriented Design Using Java, Dale Skrien, McGraw-Hill Publishing, 2008, ISBN - 0077423097, 9780077423094.

Program:	B. Tech. I.T.					Semester: **IV**	
Course: **Automata Theory**						Code: **BIT4403**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Tutorial**	**Credit**	**Hours**	**IE**	**MTE**	**ETE**	**Total**
3	1	4	4	20	30	50	100

Prior knowledge of:

1. Logic and Set Theory.
2. Computer Programming and Problem Solving

is essential

Course Objectives:

1. To understand the basics of automata theory and its operations.
2. To study various automata for different types of language classes.
3. To understand the fundamentals of decidability.

Course Outcomes:

After learning the course, the students will be able to:

1. Construct finite state machines to solve problems in computing.
2. Write regular expressions for the formal languages.
3. Construct grammar for the formal languages.
4. Design push down automata for formal languages.
5. Construct Turing machine for formal languages.
6. Explain the decidability and decidability problems.

Detailed Syllabus		
Unit	**Description**	**Duration (H)**
1.	**Finite State Machines (FSM)** Basic Concepts: Symbols, strings, formal and natural languages. FSM without output: definition and construction-Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA) with and without epsilon. FSM with output: Definition and construction of Moore and Mealy machines, applications of automata.	10
2.	**Regular Expressions** Definition of Regular Expressions (RE), Construction of regular expression of the given language, construction of language from the RE, construction of FA from the given RE using direct method, conversion of NFA to DFA, minimization of DFA, conversion of FA to RE using Arden's theorem.	9
3.	**Context Free Grammar and Languages** Introduction, formal definition of grammar, derivation process, Context Free Grammar (CFG), ambiguous CFG, removal of ambiguity, simplification of CFG, normal forms, Chomsky hierarchy, regular grammar.	9
4.	**Pushdown Automata** Introduction and definition of Push Down Automata (PDA), Construction of deterministic and non-deterministic PDA.	7
5.	**Turing Machines** Formal definition of a Turing machine (TM), Design of Turing machines, Variants of Turing machines. Comparisons of all automata.	7

6.	**Decidability** Decidability problems, recursive languages and recursively enumerable languages, decidability problems related to regular languages and context free grammar, universal Turing machine, halting problem of TM, un-decidable problems, post correspondence problem, mapping reducibility.	6
	Total	**48**

Text Books:

1. Introduction to the Theory of Computation by Michael Sipser (CENGAGE Learning, 3e).
2. Theory of Computation by Vivek Kulkarni (Oxford University Press).

Reference Books:

1. Introduction to Automata Theory, Languages and Computations by Hopcroft Ullman (Pearson Education, 3e).
2. Theory of Computation: A Problem-Solving Approach by Kavi Mahesh (Wiley-India).
3. Theory of Computation video lectures by Portland State University: Prof. Harry Porter; www.cs.pdx/~harry
4. Introduction to Computer Theory by Daniel I. A. Cohen (Wiley-India).
5. Theory of Computer Science (Automata, Languages and Computation) by K.L.P Mishra, N. Chandrasekaran (Prentice Hall India, 2e).

Program:	**B. Tech. (IT)**					Semester :**IV**	
Course :**Microprocessor & Microcontroller Laboratory**						Code :**BIT4302**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credit**	**TW**	**OR**	**PR**	**Total**
-	2	-	1	25	-	25	50

Course Objectives:

1. To learn assembly language programming of 8086 microprocessors.
2. To learn embedded C programming and PIC18FXXXmicrocontrollers.
3. To learn interfacing of real-world input and output devices to PIC18 microcontroller.

Course Outcomes:

After learning the course, the students will be able to:

1. Execute assembly language program to perform array addition.
2. Develop assembly language program for code conversion.
3. Execute assembly language program to perform string operations
4. Apply concepts related to embedded C programming.
5. Develop embedded C program to perform array addition, block transfer, sorting operations
6. Execute interfacing of real-world input and output devices to PIC18microcontroller.

Assignment No.	**Suggested List of Assignments**
GROUP A (Any 3)	
1.	Write Assembly Language Program (ALP) to add array of N numbers stored in the memory.
2.	Write menu driven ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for i. HEX to BCD ii. BCD to HEX iii. EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. Write near procedures to complete the task.
3.	Write ALP to perform following operation on string: i. Find and display length ii. Display reverse iii. Check whether string is palindrome or not. Display proper strings to prompt the user while accepting the input and displaying the result. Write near procedures to complete the task.
4.	Write menu driven ALP to perform string manipulations. The strings to be accepted from the user is to be stored in code segment Module_1 and write FAR PROCEDURES in code segment Module_2 to perform any two of the following string operations: i. Concatenation of two strings. ii. Comparison of two strings. iii. Finding Number of occurrences of a sub-string in the given string iv. Finding number of alphabets, digits, special characters, lower & upper case alphabets, words and number of lines from the text.
GROUP B(Any 3)	

5.	Study of Embedded C programming language (Overview, syntax, One simple program like addition of two numbers).
6.	Write an Embedded C program to add array of n numbers.
7.	Write an Embedded C program to transfer elements from one location to another for following: i) Internal to internal memory transfer ii) Internal to internal memory exchange.
8.	Write an Embedded C menu driven program for : i) Multiply 8 bit number by 8 bit number ii) Divide 8 bit number by 8 bit number .
9.	Write an Embedded C program for sorting the numbers in ascending and descending order.
GROUPC (Any 1)	
10.	Write an Embedded C program to interface PIC 18FXXX with LED & blinking it using specified delay.
11.	Write an Embedded C program for Timer programming ISR based buzzer on/off.
12.	Write an Embedded C program for LCD interfacing with PIC 18FXXX.
GROUP D	
13.	Study of ARM processor.
Remark: Implementation of all assignments is encouraged.	

Reference Books:

1. Peter Abel, NiyazNizamuddin, "IBM PC Assembly Language and Programming", Pearson Education.
2. Ray Duncan, "Advanced MS DOS Programming", 2nd edition, BPB Publications.
3. Mazidi, RolinMcKinlay and Danny Causey, ‘PIC Microcontroller and Embedded Systems using Assembly and C for PIC18”, Pearson Education.

Program:	**B. Tech. I.T.**					Semester: **IV**	
Course: **Data Structures and Algorithms Laboratory**						Code: **BIT4404**	
Teaching Scheme				**Evaluation Scheme**			
Lecture	**Practical**	**Tutorial**	**Credit**	**TW**	**OR**	**PR**	**Total**
-	2	-	1	25	-	25	50

Course Objectives:

1. To apply various advanced data structures such as trees and graphs to the applications.
2. To know the basics of computational complexity analysis and various algorithm design strategies.
3. To provide a thorough knowledge of the most common algorithms and data structures.
4. Design algorithms using appropriate design techniques.

Course Outcomes:

After learning the course, the students will be able to:

1. Apply and implement algorithms to create/represent and traverse non-linear data structures such as trees, graphs.
2. Experiment with Symbol tables.
3. Explain and apply the concept of hashing in database creation and manipulation.
4. Build the various algorithmic design paradigms.
5. Apply appropriate algorithmic strategy in problem-solving.
6. Analyze space and running times of the algorithms.

Assignment No.	**Suggested List of Assignments** (Write a program in C, any 8)
1.	Construct and expression tree from postfix/prefix expression and perform recursive and non-recursive In-order, pre-order and post-order traversals.
2.	Implement binary search tree and perform following operations: a. Insert b. Delete c. Search d. Mirror image e. Display f. Display level wise (algorithmic analysis)
3.	Implement DFS and BFS on a graph.
4.	Implement Merge Sort to sort the given list of numbers using divide-and-conquer strategy and verify the time complexity.
5.	Represent any real-world graph using adjacency list /adjacency matrix find minimum spanning tree using Prim's algorithm.
6.	Store data of students with telephone no and name in the structure using hashing function for telephone number and implement chaining with and without replacement.
7.	Develop functions to insert and delete into/from a max heap.
8	Implement Knapsack problem using Greedy approach.
9.	Write a program to implement Bellman-Ford Algorithm using Dynamic Programming and verify the time complexity.

10.	Write a recursive program to find the solution of placing n-queens on chessboard so that no two queens attack each other using Backtracking.

Reference Books:

1. Thomas H Cormen and Charles E.L Leiserson, “Introduction to Algorithm”, PHI, ISBN:81-203-2141-3.2. G. A.V, PAI, “Data Structures and Algorithms “, McGraw Hill, ISBN -13: 978-0-07-066726-6.
2. Gilles Brassard, Paul Bratle, “Fundamentals of Algorithms”, Pearson, ISBN 978-81-317-1244.
3. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.
4. https://ds1-iiith.vlabs.ac.in/exp/depth-first-search/index.html
5. https://ds1-iiith.vlabs.ac.in/exp/breadth-first-search/index.html

Program:	**B. Tech. I.T.**						Semester: **IV**
Course: **Object Oriented Programming Laboratory**							Code: **BIT4405**
Teaching Scheme				**Evaluation Scheme**			
Practical	**Tutorial**	**Credit**	**Hours**	**TW**	**OR**	**PR**	**Total**
4	-	2	4	50	-	25	75

Course Objectives:

1. To apply the concepts of object oriented paradigm.
2. To design and implement models for real life problems by using object oriented programming.
3. To develop object oriented programming skills.

Course Outcomes:

After learning the course, the students will be able to:

1. Construct classes for given real life data and operations.
2. Make use of static data members and methods for given real life problem
3. Design hierarchy of classes using inheritance.
4. Apply the concept of interface to solve a real-life problem.
5. Develop an application which handles different types of exceptions
6. Build an object-oriented solution to solve a real-life problem.

Assignment No.	Suggested List of Assignments
1.	Design a class 'Complex' with data members for real and imaginary part. Provide default and Parameterized constructors. Write a program to perform arithmetic operations of two complex numbers.
2.	Design a class named weather report that holds a daily weather report with data members day_of_month, hightemp, lowtemp, amount_rain and amount_snow. Use different types of constructors to initialize the objects. Write a program to generate monthly report that displays average of each attribute.
3.	Identify commonalities and differences between Publication, Book and Magazine classes. Title, Price, Copies are common instance variables and saleCopy is common method. The differences are, Bookclass has author and order_Copies(). Magazine Class has orderQty, Current_issue() andreceive_issue().Write a program to find how many copies of the given books are ordered and display the total sale of publication.
4.	Design and develop inheritance for a given case study and identify objects and relationships in it. Employee class with Emp_name, Emp_id, Address, Mail_id, and Mobile_no as data members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP, DA 10 % of BP, HRA is12% of BP, and PF is 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.
5.	Design a base class shape with two double type values and member functions to input the data and compute_area() for calculating area of figure. Derive two classes' triangle and rectangle. Make compute_area() as abstract function and redefine this function in the derived class to suit their requirements. Write a program that accepts dimensions of triangle/rectangle and display calculated area. Implement dynamic binding for given case study.

6.	Design and develop a context for given case study and implement an interface for Vehicles Consider the example of vehicles like bicycle, car, and bike. All Vehicles have common functionalities such as Gear Change, Speed up and apply breaks. Make an interface and put all these common functionalities. Bicycle, Bike, Car classes should be implemented for all these functionalities in their own class in their own way.
7.	Design a Base class STACK and Handle runtime anomalies like Overflow when the stack is full and underflow when the stack is empty. Display error codes and messages by using appropriate try and catch block to handle the exceptions thrown
	****** Implement a Mini Project using all the concepts of Object Oriented Programming implemented using JAVA ****** The project should be built using a version control system like GitHub. Sample Mini Project Implement a Mini Project for online ticket booking system in which the customer able to book a ticket based on the availability of seats. Make the payment based on the type and number of tickets booked for the particular show. Type of tickets is Gold, Platinum, Silver. Discount will be given for booking maximum of 5 or greater number of tickets only for Gold category.

Reference Books:

1. Java The complete reference, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.9th edition, 2014. ISBN: 978-0-07-180856-9(E-book available athttp://www.sietk.org/downloads/javabook.pdf).
2. Mitsunori Ogihara, "Fundamentals of Java Programming", Springer; 2018, ISBN 978-3-319-89490-4.
3. Coursera Course on "Java Programming and Software Engineering Fundamentals Specialization" by Duke University. Available at https://www.coursera.org/specializations/java-programming).
4. Coursera Course on Java for Android by Vanderbilt University. Available at (https://www.coursera.org/learn/java-for-android?specialization=android-app-development).

Program:	**B. Tech. I.T.**					Semester: **IV**
Course: **Python Programming Lab**						Code: **BIT4911A**
Teaching Scheme				**Evaluation Scheme**		
Lecture	**Practical**	**Tutorial**	**Credit**	**IE**	**MTE**	**ETE**
-	2	-	-	-	-	-

Course Objectives:

1. To learn and understand Python programming basics and paradigm.
2. To learn and understand python looping, control statements and string manipulations
3. Students should be made familiar with the concepts of GUI controls and designing GUI Applications.
4. To learn and know the concepts of file handling, exception handling and database connectivity.

Course Outcomes:

After learning the course, the students will be able to:

1. Design and implement a program to solve a real world problem.
2. Design and implement GUI application and how to handle exceptions and files.
3. Make database connectivity in python programming language.

Assignment No.	Suggested List of Assignments
1.	Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.
2.	Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.
3.	Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following: **** ********* *******
4.	A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.
5.	Make a list, say for example this one: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5
6.	Write a Python script to concatenate following dictionaries to create a new one. Sample Dictionary: dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60} Expected Result: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
7.	Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers). i. Write a method called add which returns the sum of the attributes x and y. ii. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.

	iii. Write a static method called subtract, which takes two number parameters, b and c, and returns b - c. iv. Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y.

Reference Books:

1. "Problem Solving and Programming Concepts", Maureen Spankle, 9th edition, Pearson
2. "Head First Python- A Brain Friendly Guide", Paul Barry , 2nd Edition, O'Reilly Media, Inc.
3. "How to Solve it by Computer", R. G. Dromey, 1st edition, Pearson Education
4. "Python Programming Using Problem Solving Approach", ReemaThareja, 2nd edition Oxford University Press

Program:	**B. Tech. I.T.**					Semester: **IV**
Course: **Web Development**						Code: **BIT4911B**
Teaching Scheme				**Evaluation Scheme**		
Lecture	**Practical**	**Tutorial**	**Credit**	**IE**	**MTE**	**ETE**
-	2	-	-	-	-	-

Course Objectives:

1. To apply principles of designing for effective user interfaces.
2. To quickly develop high quality and interactive website.

Course Outcomes:

After learning the course, the students will be able to:

1. Understand different web technologies.
2. Develop static web pages using HTML.
3. Build user-friendly website using different web technologies.
4. Learn to deploy the website and access through URL

Assignment No.	Suggested List of Assignments
1.	Using HTML5 layout tags develop informative page with sections which include various images, links to other pages for navigation, make use of all possible formatting (for example font, color etc.).
2.	Create a registration form in HTML using HTML form input elements viz. textbox, text area, radio button and drop down menu, check box, submit, file and reset button. Field should contain name, address, birth-date, qualification, email, phone number, gender, comments, attach photo etc. Use HTML Form elements wherever required. Align all elements using table.
3.	Create a HTML programs using frames. Three frames displaying three different HTML files. i. Frame 01–contain examples of ordered as well as unordered list ii. Frame 02–contain examples of text formatting tags i.e. <b>, <i>, <u>, <h1>--<h6>, <p>,<sup>,<sub> iii. Frame 03 – contain example of 3 by 4 tables with data. iv. Create a hyperlink in Frame 01 that opens in Frame 03 Frame1 | Frame2 Frame3
4.	Apply CSS properties Border, margins, Padding, Navigation, dropdown list to page created in first assignment.
5.	Create form in HTML with all form elements apply form validations (e.g. Email, mobile, Pin code, Password) and Write program using Java script that checks the empty values from that form and alert back using alert function
6.	Create feedback form in PHP which contains first name, last name, address, email, comment and mobile number and store that information in database. Also create a page which displays submitted feedbacks in tabular form.

Reference Books:

1. Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP, 4th Edition by Ivan Bayross, BPB Publications. ISBN: 9788183330084.
2. Professional Word Press: Design and Development by Brad Williams, David Damstra, Hal Stern, Wrox publications Web Technologies Black Book: HTML, JavaScript, PHP, Java, JSP, XML and AJAX by Kogent Learning Solutions Inc. ISBN: 9788126554560, 8126554568.