Rashtreeya Sikshana Samithi Trust

R.V COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to VTU, Belagavi)

R.V. Vidyaniketan Post, Mysuru Road
Bengaluru – 560 059

SCHEME & SYLLABUS

3rd to 8th Semesters

Information Science and Engineering

(2016 Scheme)

Department vision

To be the hub for innovation in Information Science & Engineering through Teaching, Research, Development and Consultancy; thus make the department a well known resource centre in advanced, sustainable and inclusive technology.

Department mission

ISE1: To enable students to become responsible professionals, strong in fundamentals of information science and engineering through experiential learning.

ISE2: To bring research and entrepreneurship into class rooms by continuous design of innovative solutions through research publications and dynamic development oriented curriculum.

ISE3: To facilitate continuous interaction with the outside world through student internship, faculty consultancy, workshops, faculty development programmes, industry collaboration and association with the professional societies.

ISE4: To create a new generation of entrepreneurial problem solvers for a sustainable future through green technology with an emphasis on ethical practices, inclusive societal concerns and environment.

ISE5: To promote team work through inter-disciplinary projects, co-curricular and social activities.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO	Description
PEO1	To provide adaptive and agile skills in Information Science and Engineering needed for professional excellence / higher studies /Employment, in rapidly changing scenarios.
PEO2	To provide students a strong foundation in basic sciences and its applications to technology.
PEO3	To train students in core areas of Information science and Engineering, enabling them to analyze, design and create products and solutions for the real world problems, in the context of changing technical, financial, managerial and legal issues.
PEO4	To inculcate leadership, professional ethics, effective communication, team spirit, multi-disciplinary approach in students and an ability to relate Information Engineering issues to social and environmental context.
PEO5	To motivate students to develop passion for lifelong learning, innovation, career growth and professional achievement.

PROGRAM OUTCOMES (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialisation for the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet t h e specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling to complex engineering activities, with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with t h e society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO	Description
PSO1	Recognize and appreciate the principles of theoretical foundations, data organization, data communication, security and data analytical methods in the evolving technology
PSO2	Learn the applicability of various system softwares for the development of quality products in solving real-world problems with a focus on performance optimization
PSO3	Demonstrate the ability of team work, professional ethics, communication and documentation skills in designing and implementation of software products using the SDLC principles

Lead Society:

Program Criteria

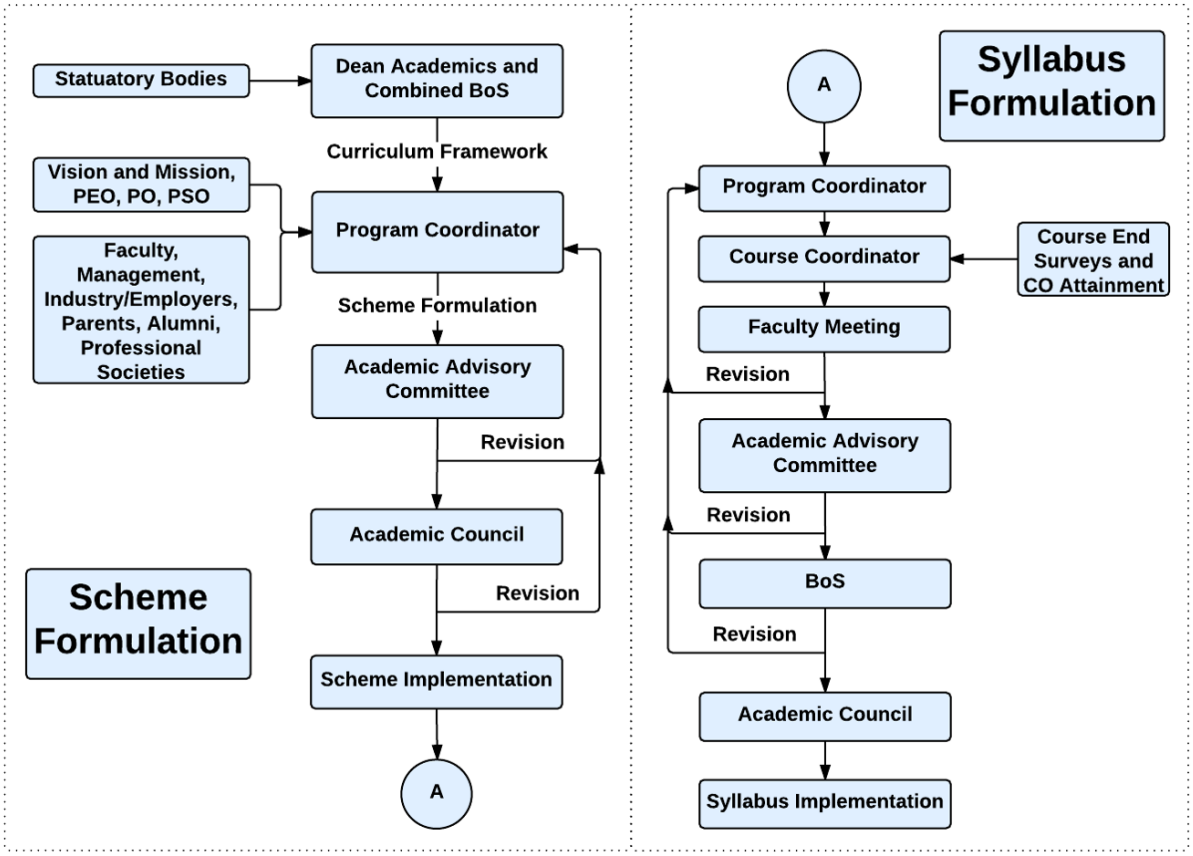
All programs seeking accreditation from the Computing Accreditation Commission of ABET must demonstrate that they satisfy all of the specific Program Criteria implied by the program title.

PROGRAM CRITERIA FOR COMPUTER SCIENCE AND SIMILARLY NAMED COMPUTING PROGRAMS

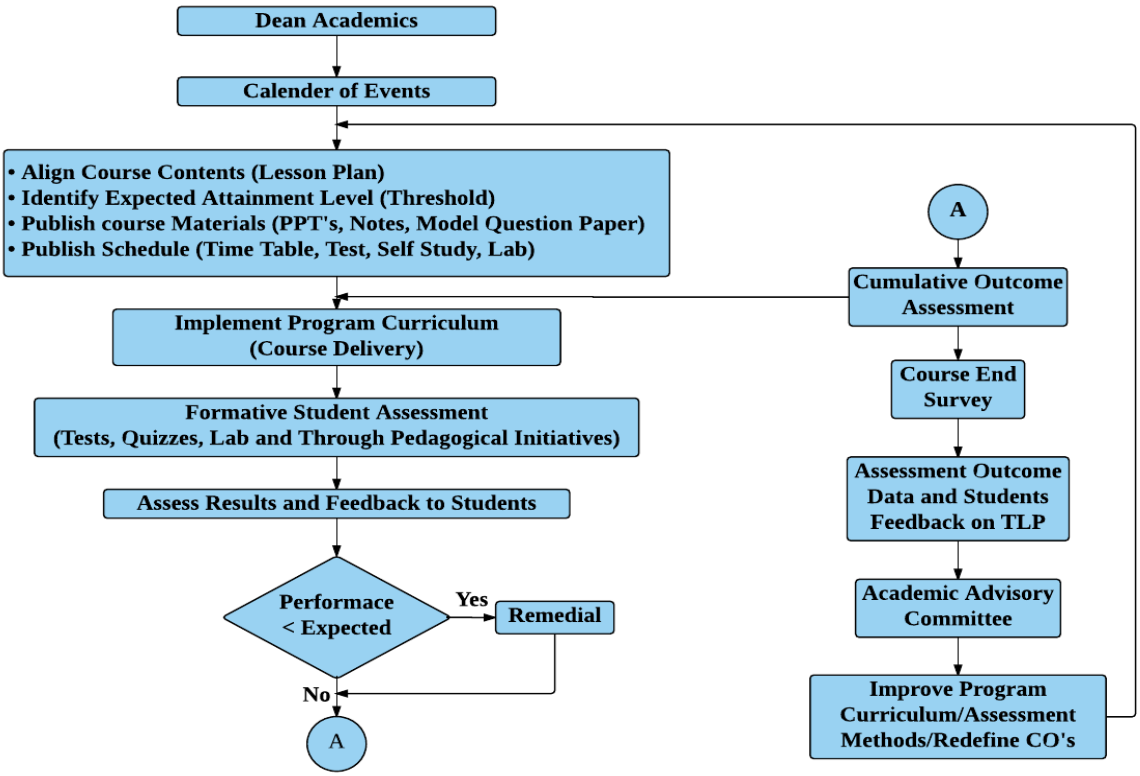
Lead Society: CSAB

Computer Science	1. Coverage of fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture.[CS] 2. An exposure to a variety of programming languages and systems.[CS] 3. Proficiency in atleast one higher-level language. [CS] 4. Advanced course work that builds on the fundamental course work to provide depth. [CS]
Information Technology	1. The core information technologies of human computer interaction, information management, programming, networking, web systems and technologies. [IT] 2. information assurance and security.[IT] 3. system administration and maintenance. [IT] 4. system integration and architecture. [IT]

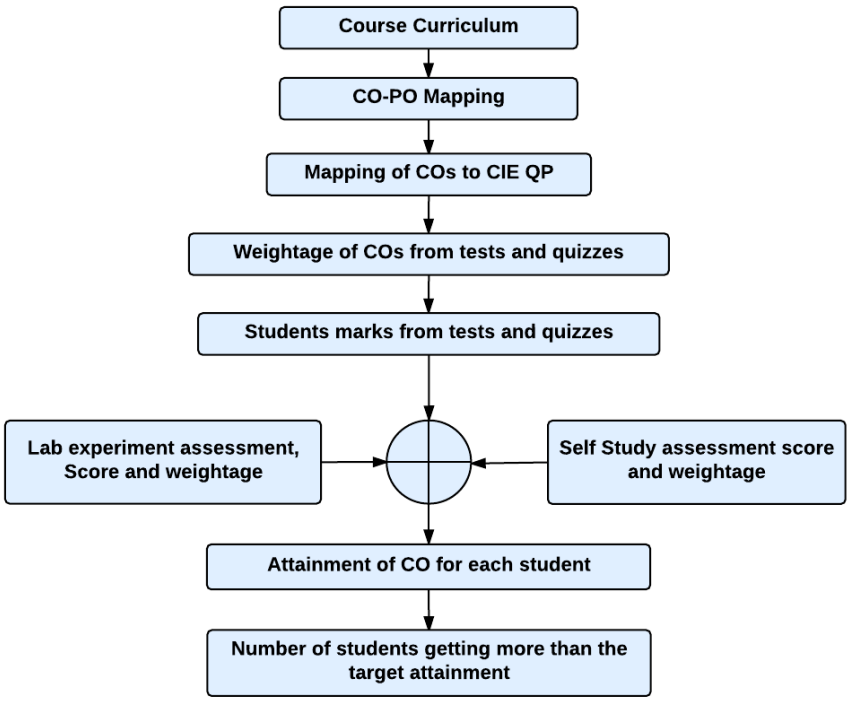
Curriculum Design Process



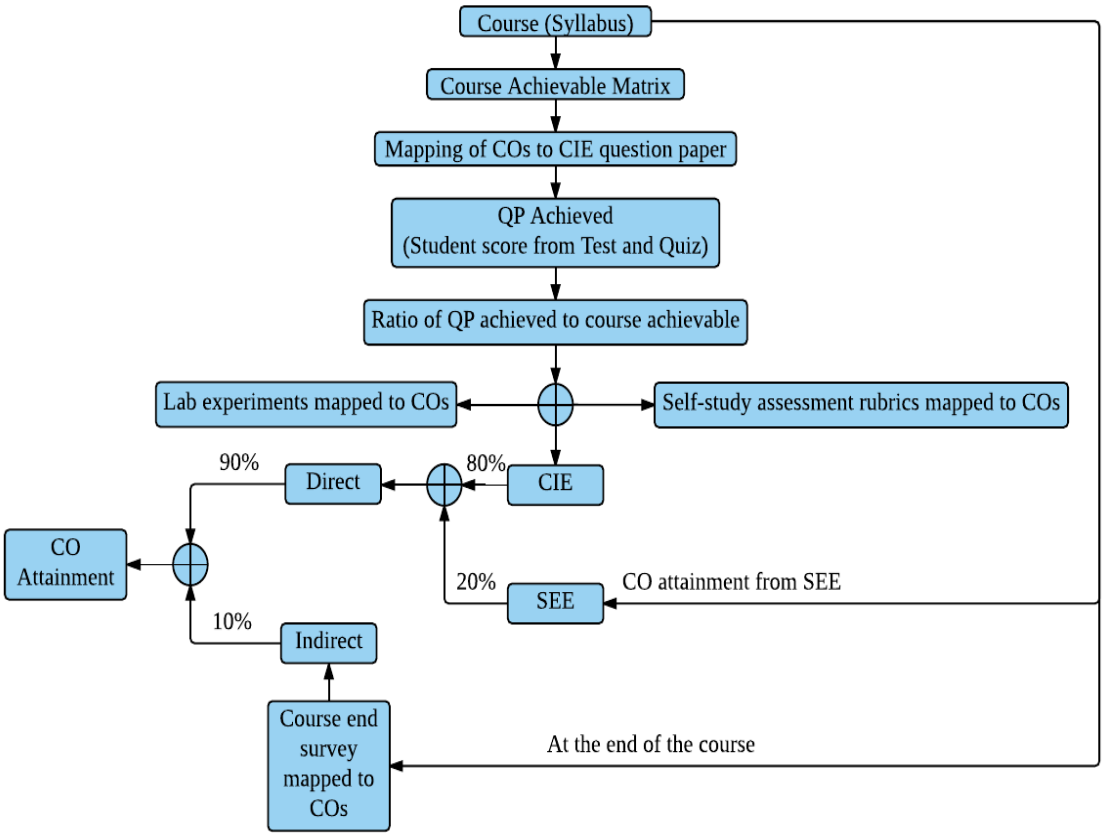
Academic Planning and Implementation



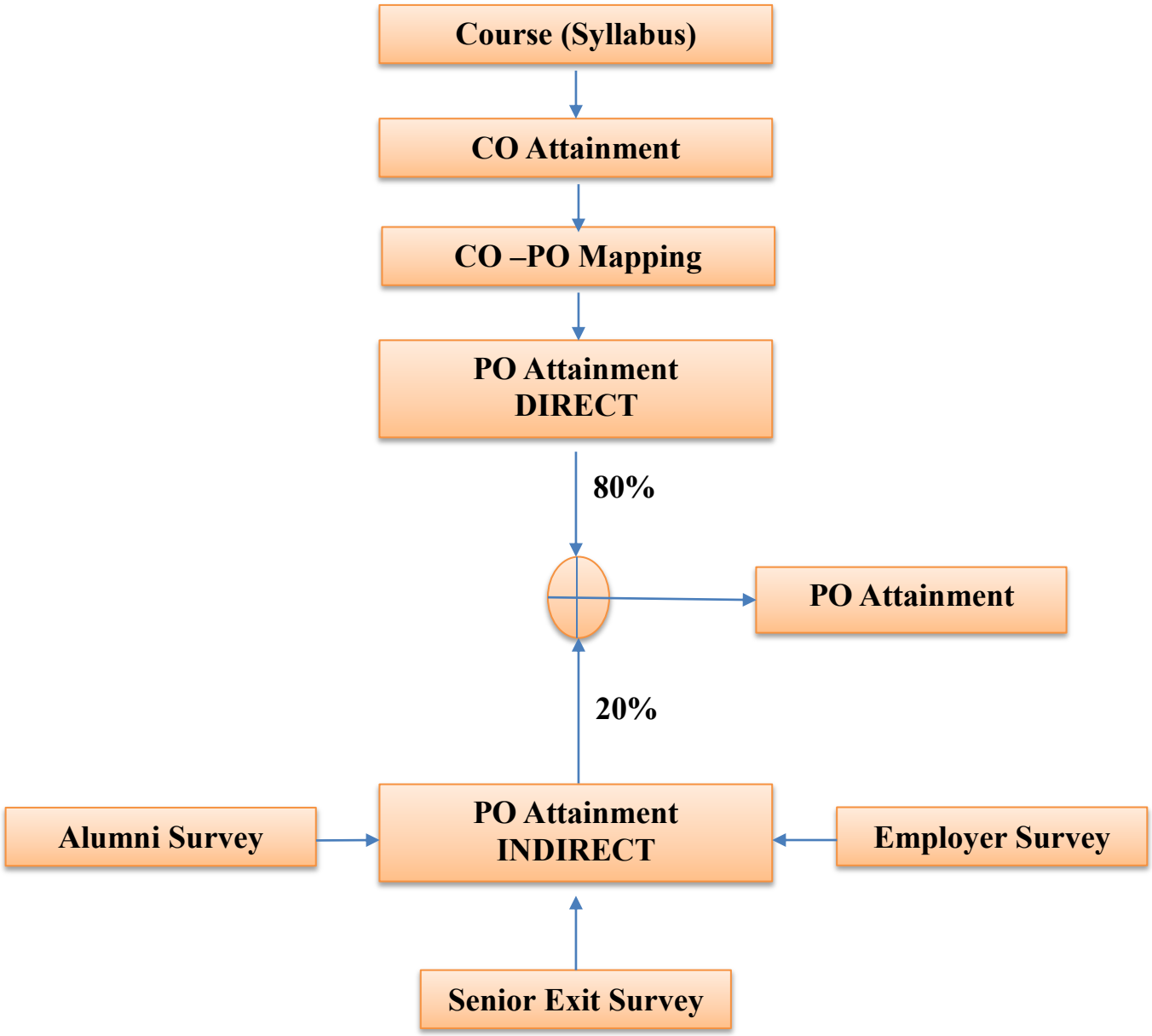
PROCESS FOR COURSE OUTCOME ATTAINMENT



Final CO Attainment Process



Program Outcome Attainment Process



Guidelines for Fixing Targets

- The target may be fixed based on last 3 years' average attainment

Credits Distribution as per UGC/VTU

Sl. No.	Category	Percentage (%)	Minimum No. of credits	2016 scheme	
				Without Mini Project	With Mini Project
1	Humanities	5-10	10	9+2	9+2
2	Basic Science	15-20	30	30	30
3	Engineering Science	15-20	30	30	30
4	Professional Core Courses (PC)	30-40	60	78+3=81 (3 credits core in place of Minor project in 7th semester)	81-3=78 (3 Credits for minor project in 7th semester)
5	Professional Elective Courses	10-15	20	20	20
6	Other Electives	5-10	10	10	10
7	Project Work	10-15	20	16+2 Major project +Tech. Seminar	16+2+3 Major project +Tech. Seminar +Mini Project
				200	200

R. V. COLLEGE OF ENGINEERING, BENGALURU – 59.

(An Autonomous Institution affiliated to VTU, Belgavi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SCHEME OF TEACHING AND EXAMINATION

THIRD SEMESTER								
Sl. No.	**Course Code**	**Course Title**	**BoS**	**Credit Allocation**				**Total Credits**
				Lecture	**Tutorial**	**Practical**	**SS (EL)**	
1	16MA31A	Fourier series,Laplace transforms and Linear algebra	Maths	3	1	0	0	4
2	16EB32	Biology for Engineers	BT	2	0	0	0	2
3	16IS33	Discrete Mathematical Structures	ISE	3	1	0	0	4
4	16IS34	Computer Organization and Architecture	ISE	4	0	0	1	5
5	16IS35	Data Structures and File Structures	ISE	3	0	1	1	5
6	16IS36	Object Oriented Programming using C++	ISE	3	0	1	1	5
7	16DCS37	Bridge Course C Programming *	CSE	2	0	0	0	0
		Total No. of Credits						25
		No. Of Hrs.					**	

*Mandatory Audit course for lateral entry diploma students

**Non contact hours

1Hr. Theory= 1 credit **2Hrs. Practical=1credit** **2Hrs. Tutorial=1 credit** **4Hrs. SS(EL) = 1 Credit**

R. V. COLLEGE OF ENGINEERING, BENGALURU – 59.

(An Autonomous Institution affiliated to VTU, Belgavi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SCHEME OF TEACHING AND EXAMINATION

FOURTH SEMESTER								
Sl. No	Course Code	Course Title	BOS	Credit Allocation				Total Credits
				Lecture	Tutorial	Practical	SS (EL)	
1	16MA41A	Graph & Probability Theory	Maths	3	1	0	0	4
2	16ET42	Environmental Technology	BT	2	0	0	0	2
3	16IS43	Operating Systems	ISE	3	1	0	0	4
4	16IS44	Design and Analysis of Algorithms	ISE	3	0	1	1	5
5	16IS45	Micro controllers and Embedded Systems	ISE	3	0	1	1	5
6	16IS46	Unix System Programming	ISE	3	0	1	1	5
7	16HSE47	Professional Practice-II (Team Work and Professional Ethics)$	HSS	0	0	0	0	1
8	16DMA48	Bridge Course Mathematics*	Maths	2	0	0	0	0
		Total No. of Credits						26
		No. Of Hrs.					12**	

*Mandatory Audit course for lateral entry diploma students **Non contact hours

$ 3 days (18 Hrs) in 3RD semester and 3 days (18 Hrs) in 4th semester, in the event of student not able to take the regular allotment, may have to complete this credit by attending other branch program. # BT, CV, CH, Chemistry will handle classes

1Hr. Theory= 1 credit 2Hrs. Practical=1credit 2Hrs. Tutorial=1 credit 4Hrs. SS(EL) = 1 Credit

Semester - III		
FOURIER SERIES, LAPLACE TRANSFORMS AND LINEAR ALGEBRA		
Course Code: 16MA31A		**CIE Marks: 100**
Hrs/Week: L:T:P:S : 3:1:0:0		**SEE Marks: 100**
Credits: 04		**SEE Duration : 3 Hrs**

Course Learning Objectives: The students will be able to	
1	Adequate exposure to basics of engineering mathematics so as to enable them to visualize the applications to engineering problems in their respective programmes.
2	To make students to understand mathematics fundamentals necessary to formulate, solve and analyze engineering problems.
3	The student should be able to analyze periodic phenomena using concept of Fourier series.
4	Apply Laplace transform technique to solve differential equation which includes the concept of convolution.
5	Use basic terminology of linear algebra in Euclidean spaces, including linear independence, spanning, basis, rank, nullity, subspaces, and linear transformations.
6	Students will become capable and eligible to participate and succeed in competitive exams like GATE, GRE.

Unit – I	
LAPLACE TRANSFORM Existence and uniqueness of Laplace Transform (LT), Transform of elementary functions, RoC. Properties of LT : Linearity, change of scale and first shifting. Transform of function multiplied by t^n, division by t, derivatives and integral. LT of periodic function, Heaviside unit step function, Unit impulse function. Heaviside shift (second shift) theorem.	**08Hrs**
Unit – II	
INVERSE LAPLACE TRANSFORM Definition, properties of inverse Laplace transform, evaluation using different methods. Convolution theorem, problems. Application to solve ordinary linear differential equations and simultaneous differential equations.	**08 Hrs**
Unit – III	
FOURIER SERIES Introduction, periodic function, even and odd functions, properties. Special waveforms - square wave, half wave rectifier, saw-tooth wave and triangular wave. Dirichlet's conditions, Euler's formula for Fourier series, Fourier series for functions of period 2L (particular cases) - problems. Half Range Fourier series-	**08 Hrs**

Construction of Half range cosine and sine series. Parseval's theorem for Root mean square value of a function. Complex form of Fourier series.	
Unit – IV	
LINEAR ALGEBRA -I Vector spaces, subspaces, Linear dependence, basis and dimension, four fundamental sub-spaces. Rank of a matrix, rank and nullity theorem, Orthonormal Bases, Gram-Schmidt process, QR-factorization.	**08Hrs**
Unit – V	
LINEAR ALGEBRA-II Linear Transformation, Geometric meaning, Matrix representation of linear transformation, Projection, reflection, rotation of linear transformation. Eigen values, Eigen vectors, Geometric meaning of Eigen values and Eigen vectors, Algebraic and Geometric multiplicity of Eigen values, Diagonalization of a Matrix, Singular Value Decomposition.	**08 Hrs**

Course outcomes: On completion of the course, the student should have acquired the ability to

1	Demonstrate the fundamental concepts in Fourier Series, Laplace transforms and Basics of Linear Algebra.
2	Identify appropriate methods to find the Fourier constants, Rank, Nullity, Orthonormal basis, Linear transformation and properties of Laplace transforms.
3	Apply the acquired knowledge to construct the Half range Fourier series, Finding Laplace transforms and Inverse Laplace transforms for some functions, Eigen values and Eigen vectors of matrix.
4	Evaluate Complex form of Fourier series, solutions of differential equations with initial and boundary condition, QR factorization, Diagonolization of matrix and Singular value decomposition.

Reference Books

1	N.P Bali & Manish Goyal, A Text Book of Engineering Mathematics, Lakshmi Publications, 7th Edition, 2010, ISBN: 978-81-7008-992-6.
2	Gilbert Strang, Linear Algebra and Its Applications, Cengage Learning India Edition, 4th Edition, 2006, ISBN: 81-315-0172-8.
3	Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons, 9th Edition, 2007, ISBN: 978-81-265-3135-6.
4	B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 40th Edition, 2007, ISBN: 81-7409-195-5.

Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation method	Course with Assignment/ Self-study
Quiz -1	05
Test -1	15
Quiz -2	05
Quiz -3	05
Test -2	15
Self-study (EL)	05
Total	100

Semester End Evaluation Theory (100)	
Part- –A Objective type questions	**20**
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1**, **UNIT-4** and **UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**
Total	**100**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**	**1**
CO2	**3**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**	**1**
CO3	**3**	**1**	**1**	**-**	**2**	**-**	**-**	**-**	**-**	**1**	**-**	**1**
CO4	**3**	**1**	**2**	**1**	**2**	**-**	**-**	**-**	**-**	**1**	**-**	**1**

Low-1Medium-2 High-3

Semester: III		
Course Title: BIOLOGY FOR ENGINEERS		
Course Code: 16EB32		**CIE Marks: 50**
Hrs/Week: L:T:P:S: 2:0:0:0		**SEE Marks: 50**
Credits: 02		**SEE Duration: 90 minutes**

Course Learning Objectives: The students will be able to	
1	To familiarize engineering students with basic biological concepts
2	To introduce students to modem biology with an emphasis on evolution of biology as a multi-disciplinary field.
3	To involve students in an interdisciplinary vision of biology and engineering
4	To gain an understanding that the design principles from nature can be translated into novel devices and structures and an appreciation for how biological systems can be engineered by human design.

UNIT-I	
Cell Types: Plant, animal and microbial cell. Stem cells: types and applications and Antibodies. Biomolecules: Carbohydrates, Proteins, Nucleic acids, Genetic code, lipids, Hormones, Vitamins, Enzymes.	**5Hrs**
UNIT-II	
Human Physiology of various systems: Digestive, Blood circulatory, Respiratory, Excretory and Nervous system. Structure and Function of sense organs- Skin, Ear, Eye, Tongue and Nose.	**5 Hrs**
UNIT-III	
Photosynthesis: Chloroplasts, Light reaction and Dark reaction. Plants as Bioinspirations: Bionic leaf and Photovoltaic cells- solar water heater and solar electricity.	**5 Hrs**
UNIT-IV	
Nature as a source of Inspiring innovation:super hydrophobic and self-cleaning surfaces - lotus leaf effect, Ultrasonography - echolocation of bats and whales, Artificial neural networks - human brain, Biosensors -natural recognition receptors, high performance fibers and flexible medical tapes - silk processing and assembly by insects and spiders, Velcro - plant burrs.	**5 Hrs**
UNIT-V	
Biomimetics for medical implants- Orthopaedic, Dental, Cardiovascular, Optical and Auditory. Artificial senses- Electronic nose and tongue.	**3 Hrs**

Course Outcomes: After completing the course, the students will be able to	
1	Remember and explain the fundamentals of biology
2	Describe the basic principles of design in biological systems.
3	Comprehend how biological principles have served as a source of inspiring innovation
4	Address the problems associated with the interaction between living and non-living materials and systems

Reference Books	
1.	Donald Voet,Charlotte W. Pratt, Judith G. Voet.," Principles of Biochemistry: International Student Version". Wiley John and Sons, 2012. ISBN: 1118092449.
2.	Yoseph Bar-Cohen, Biomimetics: Biologically Inspired technologies, CRC press, 2005, ISBN: 9780849331633
3.	Jenkins, C.H. Bioinspired Engineering, NY: Momentum press, 2012 ISBN: 97816066502259
4.	C.C.Chatterjee, Human Physiology Volume 1 (11th Edition), 2016, ISBN 10: 8123928726 / ISBN 13: 9788123928722

Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation method	Course with Assignment/ Self-study
Quiz -1	05
Test -1	15
Quiz -2	05
Quiz -3	05
Test -2	15
Self-study (EL)	05
Total	50

Semester End Evaluation (SEE) Theory (50 marks)	
Part – A	10
Objective type questions	
Part – B	40
There should be five questions from five units. Each question should be for maximum of 8 marks. The UNIT-1, UNIT-4and UNIT-5 should not have any choice. The UNIT-2 and UNIT-3 should have an internal choice. Both the questions should be of the same complexity in terms of Course outcomes and Bloom's taxonomy levels.	
Total	50

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1	1	1	-	-	1	-		1	2	-	1
CO2	1	1	2	1	-	1	-	-	1	2	-	1
CO3	1	2	2	1	-	1	-	-	1	2	-	1
CO4	2	3	3	1	2	2	1	-	1	2	-	2

Low-1Medium-2 High-3

Semester: III		
Course Title: DISCRETE MATHEMATICAL STRUCTURES		
Course Code: 16IS33		**CIE Marks : 100**
Hrs/Week: L:T:P:S 3:1:0:0		**SEE Marks : 100**
Credits:04		**SEE Duration : 3 Hrs**

Course Learning Objectives: The students will be able to	
1	Gain intense foundational introduction to fundamental concepts in discrete mathematics.
2	Interpret, identify, and apply the language associated with logical structure, sets, relations and functions, modular arithmetic.
3	Write and interpret logical statements using quantifiers.
4	Understand and apply the concepts of group and coding theory and applications.

UNIT I	
Fundamental Principles of Counting: The Rule of Sum and Product, Permutations, Combinations, The Binomial Theorem, Combinations with repetition **Mathematical Induction, Recursive Definitions** Different proof techniques, Method of mathematical induction and examples, Recursive definition and examples.	**09 Hrs**
UNIT II	
Recurrence Relations and Fundamentals of Logic: First order linear recurrence relation-Formulation problems and examples, Second order linear homogeneous recurrence relations with constant coefficients, The non-Homogeneous recurrence relations. Basic connectives and truth tables, Logical equivalence: The laws of logic, Rules of inference. Open Statement, Quantifiers, Definition and the use of Quantifiers, Definitions and the proofs of theorems.	**09 Hrs**
UNIT III	
Relations: Properties of relations, Composition of Relations, Partial Orders, Hasse Diagrams, Binary heap as a Partial order, Equivalence Relations and Partitions. **Functions:** Functions-plain, One-to-one, onto functions, Function composition and Inverse function, computational complexity, analysis of algorithms.	**09 Hrs**
UNIT IV	
Introduction to Finite Automaton and Languages: Formal language as a set, Mathematical Notations, Definitions and examples of DFA, Languages recognized by Finite Automata, Finding Equivalence classes and minimization of DFA.	**08 Hrs**
UNIT V	

Groups theory: Definition, Examples and Elementary properties, Abelian groups, Homomorphism isomorphism, cyclic groups, cosets and Lagrange's theorem. **Coding Theory:** Elementary coding theory, the hamming metric, the parity-Check and Generator Matrices.	**09 Hrs**
Expected Course Outcomes: After completing the course, the students will be able to	
1	**Reason mathematically about basic data types and structures** (such as numbers, sets, graphs, and trees) used in computer algorithms and systems
2	**Model and analyze computational processes** using analytic and combinatorial methods
3	Use abstract structures to represent discrete objects and their interrelationships
4	Apply the mathematical concepts learned to various areas of computer science
Reference Books	
1	Ralph P. Grimaldi and B V Ramana, Discrete and Combinatorial Mathematics- An Applied Introduction, Pearson Education, Asia, Fifth edition – 2007. ISBN 978-81-7758-424-0
2	J.P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata – McGraw Hill, 35^{TH} reprint 2008. ISBN 13:978-0-07-463113-3
3	Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata – McGraw Hill, Sixth Edition, Sixth reprint 2008. ISBN-(13):978-0-07-064824-1
4	C. L. Liu and D P Mohapatra, Elementary Discrete Mathematics, Tata- McGraw Hill ,Sixth Edition.ISBN:10:0-07-066913-9
5	Peter Linz, An Introduction To Formal Languages & Automata, Narosa Publishing House, VI Edition, ,2007.ISBN:978-1-4496-1552-9.

Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation method	Course with Assignment/ Self-study
Quiz -1	10
Test -1	30
Quiz -2	10
Quiz -3	10
Test -2	30
Self-study (EL)	10
Total	100

Semester End Evaluation Theory (100)	
Part- –A **Objective type questions**	**20**
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1**, **UNIT-4** and **UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**
Total	**100**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

Note: Individual faculty may adopt various methods for conducting effective quizzes and evaluate the same. The frequency of quizzes may be more than three also.

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**3**	**2**	**1**	**1**	**-**	**-**	**-**	**-**	**1**	**-**	**-**
CO2	**2**	**3**	**2**	**2**	**1**	**2**	**-**	**-**	**-**	**-**	**-**	**2**
CO3	**3**	**3**	**3**	**3**	**1**	**3**	**2**	**1**	**-**	**1**	**1**	**2**
CO4	**3**	**3**	**3**	**4**	**1**	**3**	**2**	**1**	**-**	**1**	**1**	**3**

Low-1Medium-2 High-3

Semester: III		
Course Title : Computer Organization and Architecture		
Course Code:16IS34		**CIE Marks:100**
Hrs/Week: L:T:P:S:4:0:0:1		**SEE Marks:100**
Credits:05		**SEE Duration(Theory): 3Hrs**

Course Learning Objectives: The students will be able to	
1	Understand the main components of computers and the basic principles of their operation and Interconnection Structures that realize the architecture.
2	Analyze the relationship between hardware design and instruction set architecture.
3	Explore and apply the methods for evaluating and comparing processor performance.
4	Provide a comprehensive coverage of Parallel Processing and Multi-core Architecture.

UNIT-I	
A Top-Level view of Computer Function and Interconnection: Computer Components, Computer Function, Interconnection structures, Bus Interconnection, PCI Express. **Cache Memory:** Computer Memory System Overview, Cache Memory Principles, Elements of Cache Design, Pentium 4 and ARM Cache Organization.	**09 Hrs**
UNIT-II	
Internal Memory: Semiconductor Main Memory, Error Correction. **External Memory:** Magnetic Disk, RAID, Solid State Drives, Optical Memory, Magnetic Tape. **Input/Output :** External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels and Processors, External Interface: InfiniBand.	**09 Hrs**
UNIT-III	
Computer Arithmetic: The Arithmetic and Logic Unit, Integer Representation, Integer Arithmetic, Floating-Point Representation, Floating-Point Arithmetic. **Processor Structure and Function:** Processor Organization, Register Organization, Instruction Cycle.	**09 Hrs**
UNIT-IV	
Instruction Pipelining : An Overview of Pipelining, Data Hazards: Forwarding versus Stalling , Control Hazards **Control Unit Operation :** Micro-Operations, Control of the Processor, Hardwired Implementation.	**09 Hrs**
UNIT-V	
Instruction-level parallelism And superscalar processors: Overview,	**08 Hrs**

Superscalar versus Super pipelined Constraints , Design Issues **Parallel Processing :** Multiple Processor Organizations, Symmetric Multiprocessors, Cache Coherence and the MESI Protocol, Multithreading and Chip Multiprocessors, Clusters, Non-uniform Memory Access.	

Expected Course Outcomes:After completing the course, the students will be able to	
1	Demonstrate the relationship between software and hardware and focuses on the fundamental concepts that are the basis for current computer design.
2	Describe various data representation and explain how arithmetic and logical operations are performed by computers.
3	Articulate design issues in the development of processor or other components that satisfy design requirements
4	Conceptualize , evaluate and design single and parallel processor systems to meet desired needs , within the realistic constraints specific to the field.
Reference Books	
1	William Stallings, "Computer Organization and Architecture", PHI, 9th Edition ISBN-10: 013293633X
2	David A. Patterson and John L. Hennessy, "Computer Organization and Design", Elsevier, 4th Edition, 2012, ISBN: 9780123747501.
3	Carl Hamacher, Z Vranesic& S Zaky, "Computer Organization", McGraw Hill, 5th edition, 2012, ISBN: 9781259005275
4	ShameemAkhter and Jason Roberts, "Multi-core Programming", Intel Press, 2006, ISBN: 0-9764832-4-6

Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation method	Course with Assignment/ Self-study
Quiz -1	10
Test -1	30
Quiz -2	10
Quiz -3	10
Test -2	30
Self-study (EL)	10
Total	100

Semester End Evaluation Theory (100)	
Part- –A Objective type questions	**20**
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1**, **UNIT-4** and **UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**
Total	**100**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

Note: Individual faculty may adopt various methods for conducting effective quizzes and evaluate the same. The frequency of quizzes may be more than three also.

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**1**	**2**	**2**	**2**	**-**	**-**	**-**	**1**	**3**	**1**	**1**
CO2	**3**	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**1**	**3**	**1**	**1**
CO3	**3**	**3**	**3**	**1**	**2**	**-**	**-**	**-**	**1**	**3**	**1**	**3**
CO4	**3**	**1**	**1**	**2**	**1**	**-**	**-**	**-**	**1**	**3**	**1**	**3**

Low-1Medium-2 High-3

Semester: III		
Course Title: DATA STRUCTURES AND FILE STRUCTURES		
Course Code: 16IS35		**CIE Marks: 100 + 50**
Hrs/Week: L:T:P:S 3:0:1:1		**SEE Marks: 100 + 50**
Credits:05		**SEE Duration (Theory): 3 Hrs**
		SEE Duration (Laboratory): 3Hrs

Course Learning Objectives: The students will be able to

1	Learn fundamentals of data structures and their applications essential for programming/problem solving.
2	Analyse and apply linear data structures: Stack, Queues, Lists to solve problem.
3	Analyse and apply non- linear data structures: Graphs, Trees to solve problem.
4	Learn fundamentals of file structures and its programming essentials.

UNIT I	
Performance Analysis: Mathematical Background, Running time analysis and calculations; Asymptotic notations **Abstract Data Types–**ADTs **Stacks :** Stack ADT; Stack applications: Infix to postfix conversion, Evaluation of postfix expression, Recursion	**07Hrs**
UNIT II	
Queues: Queue ADT; Circular queues; Priority queues; Queue applications: A Mazing Problem, Multiple Stacks and Queues. **Linked Lists:** List ADT; Linked implementation of Stacks, Queues; Header node; Circular linked lists; Doubly linked lists; Applications of Linked lists: Polynomial Manipulation, Multiple Precision Arithmetic.	**08Hrs**
UNIT III	
Hashing: Symboltable; Hash function; Collision resolution techniques: Open addressing, Separate chaining. **Graph:** Graph ADT; Preliminaries; Matrix and Adjacency List representation of Graphs. **Tree:** Tree ADT; Preliminaries; Binary Trees; Representation of Binary Trees; Application of Binary Tree: Evaluation of Expression, Symbol Table construction.	**08Hrs**
UNIT IV	
Search trees: Search tree ADT; Binary Search Tree; 2-3 tree; Application of Search Tree: Dictionary, Longest Prefix Matching **Heaps:** Heap ADT, Binary heap, Binomial heap	**07Hrs**
UNIT V	
Files Structures: Concepts of fields: Records and Files; Sequential, Indexed and Relative/Random File Organization; Indexing structure for index files; Hashing for direct files, B-Tree.	**06 Hrs**

Expected Course Outcomes: After completing the course, the students will be able to	
1	Comprehend how choice of data structure and file structure influences the performance of programs.
2	Analyse the running time of operations like searching, insertion, deletion traversing on various data structure and file structure.
3	Implement and demonstrate program design and implementation competence through the choice of appropriate data structure and file structure.
4	Apply appropriate data structure and file structure in solving real world problems from various domains.
Reference Books	
1	YedidyahLangsam Moshe J. Augenstein and Aaron M. Tanenbaum; Data Structures using C and C++, PHI/Pearson, 2nd Edition, 2009.
2	Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Clifford Stein; Introduction to algorithms, MIT Press, 3rd Edition, 2009
3	Jean Paul *Tremblay* and Paul G *Sorenson*; An Introduction to Data Structures with Applications, Tata McGraw Hill , 2nd Edition, 2002.
4	R.Kruse, C.L Tondo and B.Leung; Data Structures and Program Design in C++, Pearson Education, 2nd Edition, 2009.

Laboratory Component:

Students are required to implement following programs using C/C++.

Part A (Compulsory)

- Implementation of integer stack ADT using arrays
- Implementation of integer queueADT using arrays
- Implementation of integer ListADT
- Implementation of GraphADT using List
- Implementation of treeADT using List
- Implementation of basic operation on Files.
- Implementation of simple hash algorithm for files with records.

Part B

At-least one application from each of the following group.

Application of Stack

- Implementation of Infix to Postfix conversion
- Implementation of Infix to Prefix conversion
- Implementation of postfix evaluation
- Implementation of prefix evaluation

Application of Queue

- Implementation of Priority queue program using array.
- Implementation of multiple stacks and queues
- Implementation of Johnsons Algorithm
- Implementation of maze problem

Application of List

- Implementation of sparse matrix multiplication.

- Implementation of polynomials operations (addition, subtraction) using Linked List.
- Implementation of Linked Lists menu driven program (stack and queue)
- Implementation of Double ended queue using Linked Lists.

Application of Graph & Tree

- Implementation of construction of expression tree using postfix expression.
- Implementation of various operations on tree like – copying tree, mirroring a tree, counting the number of nodes in the tree, counting only leaf nodes in the tree.
- Implementation of dictionary using Binary Search Tree
- Implementation of Longest Prefix Matching.
- Implementation of Binary Heap program

Application of File Structures

- Implementation of Open addressing technique
- Implementation of separate chaining with linked list
- Implementation of B-Tree
- Implementation of secondary index on set of Records

CIE/SEE Evaluation: One Question from Part A, and relevant application from Part B has to be executed. Weightage for Part A will be 60% and Part B will be 40%.

<table>
<tr><th colspan="5">Continuous Internal Evaluation (CIE)</th></tr>
<tr><th colspan="2">(Theory – 100 Marks)</th><th colspan="2" rowspan="2">(Laboratory- 50 Marks)</th><th rowspan="2">Total (150)</th></tr>
<tr><th>Evaluation method</th><th>Course with assignment</th></tr>
<tr><td>Quiz -1</td><td>10</td><td rowspan="3">Performance of the student in the laboratory, every week</td><td rowspan="3">40</td><td rowspan="7"></td></tr>
<tr><td>Test -1</td><td>30</td></tr>
<tr><td>Quiz -2</td><td>10</td></tr>
<tr><td>Quiz -3</td><td>10</td><td rowspan="2">Test at the end of the semester</td><td rowspan="2">10</td></tr>
<tr><td>Test -2</td><td>30</td></tr>
<tr><td>Assignments</td><td>10</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Total</td><td>100</td><td>Total</td><td>50</td><td>150</td></tr>
</table>

<table>
<tr><th colspan="5">Semester End Evaluation (SEE)</th></tr>
<tr><th colspan="2">Theory (100 Marks)</th><th colspan="2">Laboratory(50 Marks)</th><th>Total (150)</th></tr>
<tr><td>Part- –A
Objective type questions</td><td>20</td><td rowspan="2">Experiment
Conduction with proper results</td><td rowspan="2">40</td><td rowspan="3"></td></tr>
<tr><td rowspan="2">Part –B
There should be five questions from five units. Each question should be for maximum of 16 Marks.

The UNIT-1, UNIT-4 and UNIT-5 should not have any choice.</td><td rowspan="2">80</td></tr>
<tr><td>Viva</td><td>10</td></tr>
</table>

The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.				
Total	**100**	**Total**	**50**	**150**

			To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**2**	**1**	**1**	**2**	**3**	**3**	**2**	**3**	**2**	**1**	**3**
CO2	**1**	**3**	**1**	**3**	**1**	**2**	**2**	**-**	**-**	**3**	**1**	**2**
CO3	**2**	**2**	**3**	**3**	**3**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
CO4	**2**	**1**	**2**	**2**	**2**	**3**	**3**	**2**	**2**	**2**	**3**	**3**

Low-1Medium-2 High-3

Semester: III		
Course Title : OBJECT ORIENTED PROGRAMMING USING C++		
Course Code:16IS36		**CIE Marks:100 + 50**
Hrs/Week: L:T:P:S:3:0:1:1		**SEE Marks:100 + 50**
Credits:05		**SEE Duration(Theory): 3Hrs** **SEE Duration(Laboratory): 3Hrs**

Course Learning Objectives: The students will be able to

1	Explain the benefits of object oriented design and understand when it is an appropriate methodology to use.
2	Apply the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design.
3	Design object oriented software solutions for small systems involving multiple classes and objects. Implement solutions in C++.
4	Test and debug C++ implementations. Apply generic programming for real time applications

UNIT-I

Introduction to Object Oriented Programming Concepts — **07 Hrs**

Principles of object oriented programming: Procedure oriented programming Vs object oriented programming, Underlying concepts of object oriented programming, Benefits and applications of object oriented programming.
The Origins of C++, A Closer Look at the I/O Operators, The bool Data Type, The C++ Headers, Namespaces, C++ programming fundamentals, , Introducing C++ Classes & objects, Constructors and Destructors, The C++ Keywords, The General Form of a C++ program, C++ I/O basics, Portability, Compiling & Linking, Pointers, Reference Types, Managing Memory in C++, Storage Classes

UNIT-II

Classes & Objects — **07 Hrs**

Discovering Classes, Interfaces, Encapsulation, Abstraction, Member Functions, Classes and Objects, Object has an interface, Structures and Classes, Unions and Classes, Friend Functions, Friend Classes, Inline Functions, Static Class Members, Static Data, Static Member Functions, Constructors and Destructors, The Scope Resolution Operator, Nested Classes, Local Classes, Passing Objects to Functions, Returning Objects, Object Assignment, Accessing Data Fields. Introduction to Object Oriented Design & The Unified Modelling Language.

UNIT-III

Inheritance and Polymorphism — **07 Hrs**

Inheritance, Access Control in derived classes, Encapsulation & protected access, Advanced operations with inheritance, Function Overloading and Default arguments, Polymorphism , operator overloading, Virtual functions and Abstract Classes

UNIT-IV	
Streams and Files, Exception Handling Streams classes, Stream Errors, Disk File I/O with streams, file pointers, error handling in file I/O with member function, overloading the extraction and insertion operators, memory as a stream object, command line arguments, and printer output. Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements, Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Understanding terminate( ) and unexpected( )	**07 Hrs**
UNIT-V	
Template Functions and Classes – Generic Programming Template Functions, compile-time Polymorphism, Template Classes, Template Linked List, Nontype Template Arguments, Setting Behavior Using Template Arguments **Standard Template Library (STL) of C++** Inline Member Function and Template, C++ Standard Library- The "String" Class, The Fundamental Containers, The Stack and Queue Adapters, Template Class "vector", Template Class "map", Template Class "list", Iterators and Algorithms The Standard Function Library and The Standard C++ Class Library	**8 Hrs**

Note : Students are advised to use SWEBOK for experiential learning available at http://www.ieeelms.com/rvce

Expected Course Outcomes:After completing the course, the students will be able to	
1	Exhibit program design and implementation competence through the choice of appropriate object oriented concept and explain the benefits of the same.
2	Design and analyze the programming applications using object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design.
3	Envision the solutions for real-time problems using Object Oriented concepts.
4	Understand and apply advanced features of C++ specifically stream I/O, templates and operator overloading which influences the performance of programs.
Reference Books	
1	Herbert Schildt, McGrawHill,The Complete Reference C++, 4th Edition, 2011, ISBN: 9780070532465.
2	Paul Deitel and Harvey Deitel, C++ How to Program, Prentice Hall, 8th Edition, 2012, ISBN: 9780132990448.
3	Cay S. Horstmann, Timothy Budd, Big C++, Wiley India (P.) Ltd, 1st Edition, 2009, ISBN: 9788126509201.
4	http://iacs-courses.seas.harvard.edu/courses/cs207/resources/TIC2Vone.pdf Bruce Eckel, Thinking in C++ - Introduction to standard C++,Pearson, Vol 1, 2nd

	Edition, 2002 , ISBN-10: 8131706613
5	Walter Savitch, "Problem Solving with C++", Addison-Wesley, 9e Global Edition, 2015, ISBN-13:9781292018249.

Laboratory Component

1. Encapsulation: Objects & Classes - C++ object, class and data abstraction fundamentals.

   The C++ programming skills that should be acquired in this lab session:

   To implement the basic principles of encapsulation, data hiding, class, object, object instance and message, use keyword public and private, use constructor and destructor, use inline function, use object packaging.

2. Encapsulation:The C++ programming abilities that should be acquired in this lab session:

- Class and arrays.
- Pointer within class.
- Pointer of the objects.
- static member variable.
- Pointer of object to another object: list and linked list examples.
- Class and strings.
- Nesting the classes.
- new and delete operators.
- this pointer.
- Default methods.

3. Inheritance: C++ object/class inheritance, extending the classes. The C++ programming abilities that should be acquired in this lab session:

- Implement inheritance concept, base class (parent class), derived class (child class).
- Implement and use pre-processor directive to avoid the multiple inclusion of the same file.
- Implement class hierarchy.
- Scope operator (::).
- protected, private and public keywords.

4. Inheritance: The C++ Inheritance programming abilities**:** Able to implement and use:

- Method vs function.
- Constructor Execution Order.
- Destructor Execution Order.

- Pointer, Array and Objects.
- Friend functions and classes, keyword friend.

5. Inheritance: Multi inheritance - C++ object/class multi inheritance, generic types. The C++ inheritance programming abilities: Able to implement and use:

- Multiple inheritances.
- Duplicated methods issue.
- Duplicated member variables issue.

6. Generic Programming: The C++ multi inheritance programming knowledge should be acquired: Able to design and implement:

- Parameterized type - Function template.
- Parameterized type - Class template.
- Generic Programming applications.

7. Polymorphism - C++ polymorphism, virtual functions. The C++ programming skills that should be acquired: Able to implement and use:

- Polymorphism concept, Virtual function, Late and early binding.
- Operators overloading.
- Functions overloading.
- C++ Formatted I/O - Standard C++ formatted input/output - cin, cout, cerr etc. The C++ formatted I/O programming skills: use various member functions for C++ formatted I/O, use various stream manipulators for C++ formatted I/O.

8. C++ File I/O - Standard C++ file input/output - read, write, create file streams. The C++ file input/output programming skills:

- use the ifstream, ofstream and fstream class objects.
- use a sequential access file – Read and Write member functions.
- use a random access file – Read and Write member functions.
- Be familiar with other file I/O member functions.

9. Storage Classes: const, volatile, static, auto, register - const, static, auto, register, volatile, mutable etc. The C++ storage classes programming abilities:

- use storage classes: auto, extern, static and register.
- use the const for variable and member function.
- use the volatile keyword.
- external and internal linkages terms.

10. C / C++ Exception Handling - Simple and structured exception handling (SEH) - try-catch-throw etc. The C & C++ programming skills that should be acquired in this session:

- use C++ exception handlings in general.
- use the assert() function.
- use try-throw-catch, structured exception handling

11. C++ Typecasting - The simple/automatic and advanced C++ type castings - simple cast, up/down/cross cast. The C and C++ programming skills that should be acquired:

- basic of type casting.
- use the automatic type casting.
- usestatic_cast, const_cast, dynamic_cast and reinterpret_cast.
- use the explicit keyword.

12. C++ Namespaces - The C++ namespaces - std, using directive etc. The C++ programming abilities that should be acquired:

- use and create the namespace.
- use namespace alias, anonymous/un-named, using directive and std.
- Using C Standard Library in C++ programs (C++ wrappers).
- Understand and appreciate the Standard C++ library.

Continuous Internal Evaluation (CIE)				
(Theory – 100 Marks)		**(Laboratory- 50 Marks)**		**Total (150)**
Evaluation method	**Course with assignment**			
Quiz -1	10	Performance of the student in the laboratory, every week	40	
Test -1	30			
Quiz -2	10			
Quiz -3	10	Test at the end of the semester	10	
Test -2	30			
Self Study (EL)	10			
Total	**100**	**Total**	**50**	**150**

Semester End Evaluation (SEE)				
Theory (100 Marks)		**Laboratory(50 Marks)**		**Total (150)**
Part- –A **Objective type questions**	**20**	**Experiment** **Conduction with proper results**	40	
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1, UNIT-4 and UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**	**Viva**	10	
Total	**100**	**Total**	**50**	**150**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**2**	**3**	**2**	**1**	**-**	**-**	**-**	**2**	**1**	**1**	**-**
CO2	**2**	**3**	**3**	**1**	**1**	**-**	**-**	**-**	**2**	**2**	**2**	**2**
CO3	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**1**	**2**	**2**	**3**
CO4	**3**	**2**	**3**	**3**	**3**	**2**	**2**	**2**	**2**	**3**	**2**	**3**

Low-1Medium-2 High-3

Semester: III		
Course Title : BRIDGE COURSE C PROGRAMMING		
Course Code: 16DCS37		**CIE Marks: 100**
Hrs/Week: L:T:P:S: 2:0:0:0		**SEE Marks: 100**
Credits: 00	**Audit Course**	**SEE Duration: 03 Hrs**

Course Learning Objectives: The students will be able to	
1	Develop arithmetic reasoning and analytical skills to apply knowledge of basic concepts of programming in C.
2	Learn basic principles of problem solving through programming.
3	Write C programs using appropriate programming constructs adopted in programming.
4	Solve complex problems using C programming.

UNIT-I	
Introduction to Reasoning, Algorithms and Flowcharts Skill development – Examples related to Arithmetical Reasoning and Analytical Reasoning. Fundamentals of algorithms and flowcharts.	**02 Hrs**
Introduction to C programming Basic structure of C program, Features of C language, Character set, C tokens, Keywords and Identifiers, Constants, Variables, Data types.	**01 Hrs**
Handling Input and Output operations Reading a character, Writing a character, Formatted input/output functions, Unformatted input/output functions.	**02 Hrs**

UNIT-II	
Operators and Expressions Arithmetic operators, Relational operators, Logical Operators, Assignment operators, Increment and decrement operators, Conditional operators, Bit-wise operators, Arithmetic expressions, evaluation of expressions, Precedence of arithmetic operators, Type conversion in expressions, Operator precedence and associativity.	**02 Hrs**
Programming Constructs **Decision Making and Branching** Decision making with 'if' statement, Simple 'if' statement, the 'if…else' statement, nesting of 'if…else' statements, The 'else if' ladder, The 'switch' statement, The '*?:'* operator, The '*goto'* statement. **Decision making and looping** The while statement, the do statement, The 'for' statement, Jumps in loops**.**	**03 Hrs**

UNIT-III	
Arrays One dimensional arrays, Declaration of one dimensional arrays. Initialization of one dimensional arrays, Two dimensional arrays, Initializing two dimensional arrays.	**02 Hrs**
Character Arrays and Strings Declaring and Initializing String Variables, Reading Strings from Terminal, Writing strings to screen, Arithmetic Operations on characters, String operations using with and without String handling functions.	**02 Hrs**

UNIT-IV	
User-defined functions Need for User Defined Functions, Definition of functions, Return values and their types, Function calls, Function declaration, Category of functions, Nesting of functions, Functions with arrays, Storage classes.	**03 Hrs**
Structures and Unions Introduction, Structure definition, Declaring structure variables, Accessing	**03 Hrs**

structure members, Structure initialization, Copying and comparing structure variables, Arrays of structure, Arrays within structures, Structures and functions, Unions.	

UNIT-V	
Pointers Introduction , Accessing the address of a variable, Declaring and initializing of pointer variables, Accessing a variable using pointers, Chain of pointers, Pointer expressions, Pointer increments and scale factor, Pointers and arrays, Pointers and character strings.	**03 Hrs**
File Managements in C Basic concepts of files, Defining and opening a file, closing of a file, Input/Output operations on files.	**01 Hrs**

Expected Course Outcomes: After completing the course, the students will be able to	
1	Understand and explore the fundamental computer concepts and basic programming principles like data types, input/output functions, operators, programming constructs and user defined functions.
2	Analyze and Develop algorithmic solutions to problems.
3	Implement and Demonstrate capabilities of writing 'C' programs in optimized, robust and reusable code.
4	Apply appropriate concepts of data structures like arrays, structures, and files to implement programs for various applications.
Reference Books	
1	P. Dey, M. Ghosh, "Programming in C", Oxford University press, First Edition, 2007, ISBN (13): 9780195687910.
2	Kernighan B.W and Dennis M. Ritchie, "The C Programming Language", Second Edition, Prentice Hall, 2005, ISBN (13): 9780131101630.
3	H. Schildt, Turbo C: The Complete Reference, Mcgraw Hill Education, 4th Edition, 2000, ISBN-13: 9780070411838.
4	Yashavant P. Kanetkar, "Understanding Pointers in C", BPB publications, 3rd edition, ISBN-13: 978-8176563581.

Scheme of Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation Method	**Course with assignment**
QUIZ - 1	**10**
TEST – 1	**30**
QUIZ – 2	**10**
QUIZ – 3	**10**
TEST – 2	**30**
ASSIGNMENT	**10**
TOTAL	**100**

Scheme of Semester End Evaluation (SEE) Theory – 100 marks	
PART – A	**20**
PART- B **There should be five questions from five units. Each question should be for maximum of 16 marks.** **The UNIT-1,UNIT-4,UNIT-5 should not have choice.** **The UNIT-2 an UNIT-3 should have internal choice.** **Both the questions should be of the same complexity in terms of CO's and bloom's taxonomy level.**	**80**
Total	**100**

Semester: IV		
Course Title : GRAPH & PROBABILITY THEORY		
Course Code:16MA41A		**CIE Marks:100**
Hrs/Week: L:T:P:S:3:1:0:0		**SEE Marks:100**
Credits:04		**SEE Duration(Theory): 3Hrs**

Course Learning Objectives: The students will be able to	
1	Learn the fundamental concepts in graph theory in view of its applications in modern science.
2	Learn to understand and create mathematical proofs, including an appreciation of its significance in C S.
3	Use the concepts of Graph theory in subsequent courses in the design and analysis of algorithms, computability theory, software engineering and computer systems.
4	Apply concepts of the theory of probability in study of random phenomena, analyzing and interpreting data that involves uncertainties.

UNIT-I	
GRAPH : Introduction, basic terminology, simple graph, degree of a vertex, types of graphs, subgraphs and isomorphic graphs, operations of graphs and connectivity, Eulerian and Hamiltonian graph, shortest path problems, representation of graphs in network flows.	**08 Hrs**
UNIT-II	
TREES : Introduction, trees and their properties, types of trees, spanning tree, Cayley's theorem (with proof)-problems, minimum spanning tree.Binary tree, properties of binary trees, m-array trees. **PLANAR GRAPHS :**Definition, Euler's formula (with proof), applications and problems, Kuratowski's theorem, matching.	**08 Hrs**
UNIT-III	
COLORINGS : Introduction, coloring of graphs, vertex coloring, chromatic number, chromatic index, chromatic polynomial, chromatic partitioning, five color theorem (with proof), four color theorem (without proof). Edge coloring of graphs.	**08 Hrs**
UNIT-IV	
PROBABILITY :Baye's rule, Random Variables: Discrete and continuous, probability mass function, probability density function, cumulative density function, mean, variance, standard deviation-problems. Joint probability distributive function- Discrete and continuous, mean, covariance and correlation	**08 Hrs**
UNIT-V	
PROBABILITY DISTRIBUTIONS : Some standard discrete and continuous Distribution- Binomial, Poisson, Normal, Exponential and Geometric	**08 Hrs**

distributions.Sampling Theory: Sampling, sampling distributions, standard errors, student's t-distribution, chi-square distribution as a test of goodness of fit.	

Expected Course Outcomes: After completing the course, the students will be able to	
1	Demonstrate the knowledge of fundamental concepts in Graph theory and Probability theory.
2	Apply models of Graph theory, Probability theory respectively to solve problems of connectivity and uncertainty.
3	Analyzing graphs, trees and random phenomena occurring in real life situations using Graph theory and Probability theory respectively.
4	Interpret the models of Graph theory, Probability theory for real life and engineering problems.

Reference Books	
1	Frank Harary, "Graph Theory", Narosa Publishing House, ISBN:978-81-850-1555-2
2	Kishor S. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer applications, 2nd Edition, Wiley Publication, ISBN: 978-0-471-33341-8
3	GeirAgnarsson & Raymond Greenlaw: Graph Theory-Modeling, Applications and Algorithms. Pearson Education, 2008, ISBN - 978-81-317-1728-8.
4	Seymour Lipschutz& Marc Lars Lipson- "Theory and Problems of Probability", Schaum's Outline Series, 2nd Edition, ISBN: 0-07-118356-6.

Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation method	Course with Assignment/ Self-study
Quiz -1	10
Test -1	30
Quiz -2	10
Quiz -3	10
Test -2	30
Assignment	10
Total	100

Semester End Evaluation Theory (100)	
Part- –A **Objective type questions**	**20**
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1**, **UNIT-4** and **UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**
Total	**100**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**	**1**
CO2	**3**	**2**	**2**	**1**	**-**	**-**	**-**	**-**	**-**	**1**	**-**	**1**
CO3	**3**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**1**	**-**	**1**
CO4	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**1**	**-**	**1**

Low-1Medium-2 High-3

Semester: IV		
Course Title: Environmental Technology		
Course Code:16ET42		**CIE Marks: 50**
Hrs/Week: L:T:P:S: 2:0:0:0		**SEE Marks: 50**
Credits: 02		**SEE Duration: 90 min**

Course Learning Objectives: The students will be able to

1	Understand the various components of environment and the significance of the sustainability of healthy environment.
2	Recognize the implications of different types of the wastes produced by natural and anthropogenic activity.
3	Learn the strategies to recover the energy from the waste.
4	Design the models that help mitigate or prevent the negative impact of proposed activity on the environment.

UNIT-I	
INTRODUCTION: Environment - Components of environment, Ecosystem – Types and structure of ecosystem. Impact of agriculture, mining, transportation and anthropogenic activities on environment and their assessment in sustainable development. Environmental acts & regulations, Role of government, legal aspects, role of non governmental organizations (NGOs), environmental education & women empowerment, ISO 14000, Environmental Impact Assessment.	**5 Hrs**
UNIT-II	
ENVIRONMENTAL POLLUTION: Air, noise, land pollution, public health aspects. Global environmental issues – Population growth, urbanization, land management, water & waste water management. Air pollution – point and non point sources of air pollution, global warming, acid rain & ozone depletion and their controlling measures (particulate and gaseous contaminants). Solid waste management, e waste management & biomedical waste management – sources, characteristics & disposal methods. Concepts of Reduce, Reuse and Recycling of the wastes.	**6 Hrs**
UNIT-III	
WATER POLLUTION: Water resources – availability and quality aspects, water borne diseases & water induced diseases, heavy metals & fluoride problems in drinking water and ground water contamination. Eutrophication, advanced waste water treatment, nutrient removal. Energy – Different types of energy, conventional sources &non conventional sources of energy, solar energy, hydro electric energy, wind energy, Nuclear energy, Biomass & Biogas Fossil Fuels, Hydrogen as an alternative energy.	**5 Hrs**
UNIT-IV	
GREEN TECHNOLOGY: Green buildings, green materials, soilless cultivation (hydroponics), sustainable manuring technology, organic oriented farming, use of biofuels, carbon foot prints, Opportunities for green technology markets, carbon capture and storage.	**4 Hrs**
UNIT-V	
RESOURCE RECOVERY SYSTEM: Processing techniques, materials recovery systems, biological conversion (composting and anaerobic digestion). Thermal conversion products (combustion, incineration, gasification, pyrolysis, use of Refuse Derived Fuels).	**5 Hrs**

Course Outcomes: After completing the course, the students will be able to	
1	Identify the components of environment and exemplify the detrimental impact of anthropogenic activities on the environment.
2	Differentiate the various types of wastes and suggest appropriate safe technological methods to manage the waste.
3	Aware of different renewable energy resources and can analyse the nature of waste and propose methods to extract clean energy.
4	Adopt the appropriate recovering methods to recover the essential resources from the wastes for reuse or recycling.

Reference Books	
1.	Gilbert, M.M. 2004. Introduction to environmental engineering and science. Pearson Education. 2nd Edition, ISBN: 8129072770
2.	Howard S. Peavy, Donald R. Rowe and George Tchobanoglous. Environmental Engineering, McGraw Hill Series in water resources and Environmental Engg. 2000. ISBN: 0070491348
3.	G. Tyler Miller (Author), Scott Spoolman (Author), (2012) Environmental Science – Publisher: Brooks Cole, 15th edition, ISBN-13: 978-1305090446 ISBN-10: 130509044
4.	Vijay Kulkarni and T. V. Ramachandra Environment Management. 2009. TERI Press; ISBN: 8179931846, 9788179931844
5.	Sven Erik Jørgensen 2002. Integration of Ecosystem Theories: A Pattern Ecology & Environment; Edition 3, Springer; ISBN: 1402007558, 9781402007552

Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation method	Course with Assignment/ Self-study
Quiz -1	05
Test -1	15
Quiz -2	05
Quiz -3	05
Test -2	15
Self-study (EL)	05
Total	50

Semester End Evaluation (SEE) Theory (50 marks)	
Part – A	10
Objective type questions	
Part – B	40
There should be five questions from five units. Each question should be for maximum of 8 marks. The UNIT-1, UNIT-4 and UNIT-5 should not have any6 choice. The UNIT-2 and UNIT-3 should have an internal choice. Both the questions should be of the same complexity in terms of Cos and Bloom's taxonomy level.	
Total	50

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1	-	-	-	-	-	3	-	2	-		-
CO2	2	3	3	2	1	-	3	3	2	-	2	1
CO3	-	3	1	3	-	2	3	3	2	-	1	2
CO4	1	-	2	1	3	-	2	-	2	-		2

Low-1Medium-2 High-3

Semester: IV		
Course Title : OPERATING SYSTEMS		
Course Code:16IS43		**CIE Marks:100**
Hrs/Week: L:T:P:S:3:1:0:0		**SEE Marks:100**
Credits:04		**SEE Duration(Theory): 3Hrs**

Course Learning Objectives: The students will be able to	
1	Study the basic concepts of operating systems and understand the structure and functions of OS.
2	Learn about Processes, Threads, Scheduling algorithms and the principles of concurrency and Deadlocks.
3	Learn various memory management schemes and study I/O management and File systems.
4	Learn the basics of Linux system and perform administrative tasks on Linux Servers.

UNIT-I	
OPERATING SYSTEMS OVERVIEW :Computer System Overview-Basic Elements, Instruction Execution, Interrupts, Memory Hierarchy, Cache Memory, Direct Memory Access, Multiprocessor and Multicore Organization. Operating system overview-objectives and functions, Evolution of Operating System.- Computer System Organization- Operating System Structure and Operations- System Calls, System Programs, OS Generation and System Boot.	**09 Hrs**
UNIT-II	
PROCESS MANAGEMENT Processes-Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication; Threads-Overview, Multicore Programming, Multithreading Models; Windows 7 – Thread and SMP Management. Process Synchronization – Critical Section Problem, Mutex Locks, Semophores, Monitors; CPU Scheduling and Deadlocks.	**09 Hrs**
UNIT-III	
DEADLOCKS: Definition, Deadlock characteristics , Deadlock Prevention, Deadlock Avoidance :banker's algorithm, Deadlock detection and Recovery. **STORAGE MANAGEMENT** Main Memory-Contiguous Memory Allocation, Segmentation, Paging, 32 and 64 bit architecture Examples; Virtual Memory- Demand Paging, Page Replacement, Allocation, Thrashing; Allocating Kernel Memory, OS Examples.	**09 Hrs**
UNIT-IV	

I/O SYSTEMS Mass Storage Structure- Overview, Disk Scheduling and Management; File System Storage-File Concepts, Directory and Disk Structure, Sharing and Protection; File System Implementation- File System Structure, Directory Structure, Allocation Methods, Free Space Management, I/O Systems. **Security & Protection** Security Environment, Design Principles Of Security, User Authentication, Protection Mechanism : Protection Domain, Access Control List	**09 Hrs**
UNIT-V	
CASE STUDY Linux System- Basic Concepts; System Administration-Requirements for Linux System Administrator, Setting up a LINUX Multifunction Server, Domain Name System, Setting Up Local Network Services; Virtualization- Basic Concepts, Setting Up Xen, VMware on Linux Host and Adding Guest OS.	**08 Hrs**

Expected Course Outcomes: After completing the course, the students will be able to	
1	Design various Scheduling algorithms and apply the principles of concurrency.
2	Design deadlock, prevention and avoidance algorithms and compare various memory management schemes.
3	Design and Implement a prototype file systems
4	Perform administrative tasks on Linux Servers.

Reference Books	
1	Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", John Wiley and Sons Inc., 9th Edition, 2012.
2	William Stallings, "Operating Systems – Internals and Design Principles", Prentice Hall, 8th Edition, 2015.
3	Andrew S. Tanenbaum, "Modern Operating Systems", Addison Wesley, Fourth Edition, 2015.
4	D M Dhamdhere, "Operating Systems: A Concept-Based Approach", Tata McGraw-Hill Education, Second Edition, 2007.

Continuous Internal Evaluation (CIE) (Theory – 100 Marks)	
Evaluation method	Course with Assignment/ Self-study
Quiz -1	10
Test -1	30
Quiz -2	10
Quiz -3	10
Test -2	30
Assignment	10
Total	100

Semester End Evaluation Theory (100)	
Part- –A Objective type questions	**20**
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1**, **UNIT-4** and **UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**
Total	**100**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	-	2	-	-	1	1	-	1	-	3
CO2	2	3	1	-	3	-	1	-	-	-	-	2
CO3	2	1	-	1	2	-	2	-	-	3	3	1
CO4	1	2	3	-	-	-	-	1	-	-	1	1

Low-1Medium-2 High-3

Semester: IV		
Course Title : Design and Analysis of Algorithms		
Course Code:16IS44		**CIE Marks: 100 + 50**
Hrs/Week: L:T:P:S: 3:0:1:1		**SEE Marks: 100 + 50**
Credits:05		**SEE Duration(Theory): 3 Hrs** **SEE Duration(Laboratory): 3 Hrs**

Course Learning Objectives: The students will be able to	
1	Learn a mathematical model to find complexity of algorithms
2	To learn different algorithm design techniques and the algorithms that employs these techniques
3	Analyze the efficiency of algorithms using time and space complexity theory
4	Understand different algorithmic design strategies

UNIT-I	
Fundamentals of Algorithm Analysis : Definition of algorithm, Algorithmic Problem Solving, Framework for Analysis of algorithm efficiency **Asymptotic Notations :** Basic Efficiency Classes Mathematical Analysis of Non recursive algorithms, Mathematical Analysis of Recursive Algorithms, Empirical analysis of algorithms. **Divide and Conquer:** Introduction to Divide and Conquer, Master Theorem, Merge sort	**07 Hrs**
UNIT-II	
Divide and Conquer: Quick sort, Multiplication of Large Integers **Decrease and conquer :** Depth First Search(DFS) and Breadth First Search(BFS) with applications, Topological Sorting, Fake coin Problem, Computing a median and selection problem **Transform and Conquer :** Introduction, Presorting, Balanced Search Trees, Heap sort	**08 Hrs**
UNIT-III	
Dynamic Programming(DP): Floyd's Algorithm, The Knapsack Problem – Brute force method, bottom-up DP method and Memory Functions. **Greedy Technique:** Introduction, Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees, Iterative improvement – Introduction, The maximum-flow problem, maximum matching in bipartite graphs, Lower-Bound Arguments, Decision Trees	**08 Hrs**
UNIT-IV	
Space and Time tradeoff – Naïve method of string matching, Boyer-Moore and Horspool's string matching algorithms, **Limitations of algorithm power** – Lower bound arguments, Decision Trees, P, NP and NP-Complete Problems,	**07Hrs**
UNIT-V	
Coping with limitations of algorithm Power - Backtracking(BT) :N-queens and Subset-Sum problems, **Branch-and-Bound:** Assignment problem, Knapsack problem and travelling salesman problem	**06Hrs**

Note : Students are advised to use SWEBOK for experiential learning available athttp://www.ieeelms.com/rvce

Course Outcomes: After completing the course, the students will be able to	
1	**Analyze** the running time of the basic algorithms for those classic problems in various domains
2	**Apply** and implement learned algorithm design techniques and data structures to solve problems
3	**Develop** algorithms for various computing problems
4	**Identify** the limitations of algorithms in problem solving
Reference Books	
1	LevitinA.,IntroductiontoTheDesignandAnalysisofAlgorithms,PearsonEducation,2003,ISBN:9780201743951
2	CormenT.H.,LeisersonC.E.,RivestR.L.,SteinC.,IntroductiontoAlgorithms,3rdedition,PHI2010,ISBN:9780262033848
3	HorowitzE.,SahaniS.,RajasekharanS.,ComputerAlgorithms,GalgotiaPublications,2001,ISBN:9780716783169
4	MarkAllenWeiss,DatastructuresandalgorithmanalysisinC++,PearsonEducation,2003, ISBN:032144146

Lab

GeneralGuideline

1.GOTOstatementsarenotallowed

2.Noglobaldeclarationsallowed

3.Prototypeforeachuser-defined-functionmustbeprovidedbeforemain

4.main should be the first function in any program

5.Studentsareencouragedtouseuser-defined-headerfiles

6.Programsmustbeindentedappropriately

7.Studentsarerequiredtobringonlythealgorithmsinthedatasheet

Implement the following using C++Language.

1. Write a program to sort a given set of elements using Merge sort method and find the time required to sort the elements.
2. Write a program to sort a given set of elements using Quick sort method and find the time required to sort the elements.
3. Write a program to print all the nodes reachable from a given starting node in a graph using Depth First Search method. Also check connectivity of the graph. If the graph is not connected, display the number of components in the graph.
4. Write a program to obtain the Topological ordering of vertices in a given digraph using
    a) Vertices deletion method
    b) DFS method

5. Write a program to print all the nodes reachable from a given starting node in a graph using Breadth First Search method. Also check connectivity of the graph. If the graph is not connected, display the number of components in the graph.

6. Write a program to sort a given set of elements using Heap sort method. Find the time complexity.

7a. Write a program to implement Horspool algorithm for String Matching

7b. Write a program to implement all pair shortest paths problem using Floyd's algorithm.

8. Write a program to implement 0/1 Knapsack problem using dynamic programming

9. Write a program to find Minimum cost spanning tree of a given undirected graph using Prim's algorithm.

10. Write a program to find Minimum cost spanning tree of a given undirected graph using Kruskal's algorithm.

11. Write a program to find the shortest path using Dijkstra's algorithm for a weighted connected graph.

12. Write a program to implement Subset-Sum problem using Back Tracking

13. Write a program to implement Assignment Problem using branch and bound algorithm

14. Write a program to implement n-queens problem

Continuous Internal Evaluation (CIE)				
(Theory – 100 Marks)		**(Laboratory- 50 Marks)**		**Total (150)**
Evaluation method	**Course with assignment**			
Quiz -1	10	Performance of the student in the laboratory, every week	40	
Test -1	30			
Quiz -2	10			
Quiz -3	10	Test at the end of the semester	10	
Test -2	30			
Assignments	10			
Total	**100**	**Total**	**50**	**150**

Semester End Evaluation (SEE)				
Theory (100 Marks)		**Laboratory(50 Marks)**		**Total (150)**
Part- –A **Objective type questions**	**20**	**Experiment Conduction with proper results**	40	
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1, UNIT-4 and UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**	**Viva**	10	
Total	**100**	**Total**	**50**	**150**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO MAPPING												
CO/PO	**PO1**	**PO2**	**PO 3**	**PO4**	**PO5**	**PO6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	3	3	2	2	-	-	-	2	2	-	-
CO2	3	3	2	2	-	-	-	-	3	-	3	3
CO3	3	3	3	2	-	-	-	-	-	2	-	3
CO4	3	3	2	2	-	-	-	-	3	-	3	2

Low-1Medium-2 High-3

Semester: IV		
Course Title : MICRO CONTROLLERS AND EMBEDDED SYSTEMS		
Course Code:16IS45		**CIE Marks:100 + 50**
Hrs/Week: L:T:P:S: 3:0:1:1		**SEE Marks:100 + 50**
Credits:05		**SEE Duration(Theory): 3Hrs** **SEE Duration(Laboratory): 3Hrs**

Course Learning Objectives: The students will be able to

1	Provide the student with the basic understanding of microcontroller and embedded systems design.
2	Learn the addressing modes, instructions, assembler directives and develop the flow chart, algorithms to solve problems.
3	Use of subroutines, multi-segments, macros, interrupts, procedures, stacks programs in applications
4	Develop embedded C programs for microcontrollers and run on the simulator, target board and various interfaced hardware devices.

UNIT-I

Introduction to Microcontrollers & Architecture Intel 8051 Microcontroller Introduction, Microprocessor Versus Microcontroller, 8051 Block Diagram, Registers, Flags &PSW, Memory Organization: Program & Data Memory, Stack structure, Addressing Modes, Data transfer Instruction's, Structure of ALP , Working with Keil Software Tools to develop, simulate& debug ALP & embedded C programs, Assembler Directives. **Case Study: Study the architecture of 8051 variant - NXPs 89V51RD2**	**08 Hrs**

UNIT-II

Intel 8051 Instruction Set & Assembly Language Programming Study of Instruction set: Arithmetic, Logic, Jump, Loop & Call Instructions, Assembly Language Programming, Procedures, Working & Programming of Timers/Counters**,** Interrupts& ISR Programs, Writing Delay programs using Instructions &Timers. **Case Study: Comparison of Applications built using: Programmed I/O & Interrupt I/O**	**08 Hrs**

UNIT-III

Intel 8051 Interfacing & Applications Signal/Pin Descriptions, I/O Ports, Interfacing & Programming(using ALP/Embedded C) with LEDs, Switches, Seven segment displays, LCD, Matrix	**08 Hrs**

Keypad, Parallel ADC (ADC0804) , DAC (DAC0800), Stepper motor, DC Motor, Programming serial port of 8051, Communication of 8051 with the PC using serial port. **Case Study: Building PC based Embedded System Using 8051 kit & RS-232**	
UNIT-IV	
Introduction to Embedded Systems & ARM Processor/Controller Definition, Desirable Features & General Characteristics of embedded systems, Embedded Systems Vs General Computing Systems, Model of an Embedded System, Classification of Embedded Systems. History of the ARM Processor, ARM Architecture, Interrupt vector table, brief overview of ARM Instruction Set & Simple ALP Programs, Current Trends **Case Study: Example of embedded system– RFID**	**08 Hrs**
UNIT-V	
ARM7 MCU LPC2148 – Architecture & Peripheral Programming using embedded C History of the ARM Processor, ARM Architecture, Interrupt vector table, The internal architecture of LPC 2148 (a typical and popular ARM7 MCU) – Features of the LPC 214X Family, Peripherals and Programming : GPIO, Timers, PWM , UART, SSP units, **Case Study: Building Data Acquisition System using MCB 2140 compatible board**.	**08 Hrs**

Laboratory Component:

PART-A

Set 1:

a) 8051 ALP programs to perform block data transfer and searching operations

b) 8051 ALP/Embedded C to Interface Logical Controller and perform:

a. Write an ALP to read the status of 8 inputs bits from 8bit switch and display 'FF' if it is even parity otherwise display 00. Also display number of 1's in the input data on the LED outputs, using interface module.

b. Write an ALP to read the status of two 8-bit inputs (X and Y) and display the result X*Y using the interface module

c. Write an ALP to implement BCD Up/Down counters

Set 2:

a) 8051 ALP programs to perform Arithmetic (addn/subn/mult/divn operations)

b) 8051 ALP/Embedded C to Interface Seven Segment Display and perform:

a. Write a C program to display messages "FIRE" & "HELP" on 4 digit seven segment display alternately with a suitable delay.

b. Write a C program to display the given number on the seven segment display using look up table

Set 3:

a) 8051 ALP programs to perform number conversions, binary to BCD,binary to ASCII

b) 8051 ALP/Embedded C to Interface Stepper Motor Module and perform:

a. Write an Embedded C program to rotate stepper motor in clockwise direction for "M" steps, anti-clock wisedirection for "N" steps

b. Rotate the Stepper Motor, for the given RPM

Set 4:

a) 8051 ALP programs to compute average & maximum/minimum values

b) 8051 ALP/Embedded C to Interface DAC Module and perform:

a. Write an Embedded C program to generate without rectification / full rectified/ half rectified sine waveform using DAC module

b. Write the program to generate square waveform for the given frequency

c. Generate PWM wave on pin P0.1 to control speed of DC motor. Control the duty cycle by analog input.

Set 5:

a) 8051 ALP programs to perform sortingoperations

b) 8051 ALP/Embedded C to Interface Keyboard Module and perform:

a. Write an Embedded C program to interface 4 X 4 matrix keyboard using lookup table and display the key pressed on the Terminal

b. Interface an LCD Module and display the temperature read from ADC Module.

Set 6:

a) To write and simulate ARM assembly language programs for data transfer, arithmetic and logical operations

b) Interface Graphics LCD and I2C device to ARM Microcontroller LPC 2148 / 1768 and write the suitable embedded C program

Mini Projects :

1. Design & Development of PC based Embedded system using 8051 Kit, incorporating Application Development on both PC & Microcontroller
2. Design & Development of LPC 2148/1768 based Data Acquisition System

Course Outcomes: After completing the course, the students will be able to	
1	Acquire the knowledge of architecture of Microprocessors and Microcontrollers.
2	Develop skill in simple program writing for micro controllers assembly level language and Embedded C.
3	Apply acquired knowledge to design for interface and programming.
4	Analyze the design and implement for applications.
Reference Books	
1	Muhammad Ali Mazidi, Janice GillispieMazidi , Rolin D. McKinlay, The 8051 Microcontroller &Embedded Systems (Using Assembly & C), New Edition, Prentice Hall (Pearson) 2014, ISBN-13-978-1-292-02657-2.
2	Kenneth J. Ayala, "The 8051 Microcontroller Architecture, Programming &Applications", Thomson Learning, 2nd Edition, 2007.
3	Lyla B. Das, "Embedded Systems – An integrated approach", Pearson Education,

	First Impression 2013, ISBN- 978-81-317-8766-3
4	ARM system developers guide, Andrew N Sloss, Dominic Symes and Chris Wright, Elsevier,Morgan Kaufman publishers, 2008.
5	Raj Kamal, "Embedded Systems, Architecture, Programming and Design", Tata McGraw-Hill, Second Edition-Reprint 2011,ISBN-978-0-07-066764-8
6	Michael J.Pont, "Embedded C", Pearson Education, Reprint 2013, ISBN-978-81-317-1589-5

Continuous Internal Evaluation (CIE)				
(Theory – 100 Marks)		**(Laboratory- 50 Marks)**		**Total (150)**
Evaluation method	**Course with assignment**			
Quiz -1	10	Performance of the student in the laboratory, every week	40	
Test -1	30			
Quiz -2	10			
Quiz -3	10	Test at the end of the semester	10	
Test -2	30			
Assignments	10			
Total	**100**	**Total**	**50**	**150**

Semester End Evaluation (SEE)				
Theory (100 Marks)		**Laboratory(50 Marks)**		**Total (150)**
Part- –A **Objective type questions**	**20**	**Experiment** **Conduction with proper results**	40	
Part –B There should be five questions from five units. Each question should be for maximum of 16 Marks. The **UNIT-1, UNIT-4 and UNIT-5** should not have any choice. The **UNIT-2 and UNIT-3** should have an internal choice. Both the questions should be of the same complexity in terms of COs and Bloom's taxonomy level.	**80**	**Viva**	10	
Total	**100**	**Total**	**50**	**150**

	What		To whom	Frequency of conduction	Max Marks	Evidence	Contribution to Course Outcome		
Direct Assessment Methods	CIE	Quiz	Students	Three	30	Answer Scripts	80 %	100 %	90 %
		Test		Two	60/50				
		Assignment/Self-study		2 phases	10/20	Reports / Record Books			
		Laboratory		Weekly	50				
	SEE	Semester End Examination		End of every semester Consisting of Part-A and Part-B	100	Answer Scripts	20 %		
		Semester End Laboratory		End of every semester laboratory	50				
Indirect Assessment methods	Course End Survey		Students	End of course		Questionnaire Based on COs	10%		

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	1	3	2	3	1	1	-	3	2	2	2
CO2	3	2	3	2	3	-	-	-	3	2	2	2
CO3	2	2	3	2	3	1	1	-	3	2	2	2
CO4	2	2	3	2	3	1	1	-	3	2	2	2

Low-1Medium-2 High-3