Program Structure for Second Year Computer Engineering
Semester- III-Credit Scheme

Course Code	Course Name	Teaching Scheme (Hrs.) TH – P – TUT	Total (Hrs.)	Credits Assigned TH – P – TUT	Total Credits	Course Category
CEC301	Applications of Mathematics in Engineering-I	3– 0 – 1	04	3 – 0 – 1	04	BS
CEC302	Discrete Structures and Graph Theory	2– 0 – 0	02	2 – 0 – 0	02	ES
CEC303	Data Structure	3– 0 – 0	03	3 – 0 – 0	03	PC
CEC304	Digital Logic & Computer Architecture	3 – 0 – 0	03	3 – 0 – 0	03	PC
CEC305	Computer Graphics	3 – 0 – 0	03	3 – 0 – 0	03	PC
CEL303	Data Structure Lab	0 – 2 – 0	02	0 – 1 – 0	01	PC
CEL304	Digital Logic & Computer Architecture Lab	0 – 2 – 0	02	0 – 1 – 0	01	PC
CEL305	Computer Graphics Lab	0 – 2 – 0	02	0 – 1 – 0	01	PC
CEPR31	Project Based Learning: Mini Project Lab-I	0 – 2 – 0	02*	0 – 1 – 0	01	PBL
CEXS33	Skill Based Learning: Object Oriented Programming with Java (SAT-III)	0 – 2 – 0	02$	0 – 1 – 0	01	SAT
CEXS34	Skill Based Learning: (SAT-IV) (Interdisciplinary Informatics)	0 – 2 – 0	02$	0 – 1 – 0	01	SAT
INT31	Internship-II	2 to 3 Weeks		--	-	INT
Total		**14– 12 –1**	**27**	**14 - 6– 1**	**21**	**--**

*Load of learner, not the faculty $SAT Hours are under Practical head but can be taken as Theory or Practical or both as per the need.

Mini Project I and II: Students can form groups with minimum 2 (Two) andmaximum 4(Four) **Faculty Load:** 1 hour per week per four groups

Semester- III-Examination Scheme

Course Code	Course Name	Marks										
		CA				ESE	ESE duration(Hrs)	TW	O	P	P&O	Total
		T-1	T-2	Average (T-1 & T-2)	IA							
CEC301	Applications of Mathematics in Engineering-I	30	30	30	10	60	2.30	25	--	--	--	125
CEC302	Discrete Structures and Graph Theory	20	20	20	10	45	2	--	--	--	--	75
CEC303	Data Structure	30	30	30	10	60	2.30	--	--	--	--	100
CEC304	Digital Logic & Computer Architecture	30	30	30	10	60	2.30	--	--	--	--	100
CEC305	Computer Graphics	30	30	30	10	60	2.30	--	--	--	--	100
CEL303	Data Structure Lab	--	--	--	--	--	--	25	--	--	25	50
CEL304	Digital Logic & Computer Architecture Lab		--	--	--	--	--	25	--	--	--	25
CEL305	Computer Graphics Lab	--	--	--	--	--	--	25	--	--	25	50
CEPR31	Project Based Learning: Mini Project Lab-I	--	--	--	--	--	--	25	--	--	25	50
CEXS33	Skill Based Learning: Object Oriented Programming with Java (SAT-III)	--	--	--	--	--	--	25	--	--	--	25
CEXS34	Skill Based Learning (SAT-IV) (Foreign and Indian Regional Languages-I)	--	--	--	--	--	--	25	--	--	--	25
INT31	Internship-II	--	--	--	--	--		--	--	--	--	--
Total		**140**	**140**	**140**	**50**	**285**	**--**	**175**	**--**	**--**	**75**	**725**

Course Code	Course Name	Credits Assigned			
		TH	P	TUT	Total
CEC301	Applications of Mathematics in Engineering-I	03	-	01	04

Prerequisites:	1. Engineering Mathematics-I 2. Engineering Mathematics-II
Course Objectives:	1. To learn the Laplace Transform, Inverse Laplace Transform of various functions, its applications. 2. To understand the concept of Fourier Series, its complex form and enhance the problem-solving skills. 3. To understand the concept of complex variables, C-R equations with applications. 4. To understand the basic techniques of statistics like correlation, regression, and curve fitting for data analysis, Machine learning and AI. 5. To understand some advanced topics of probability, random variables with their distributions and expectations.
Course Outcomes:	**Upon completion of the course, the learners will be able to..** 1. Solve the real integrals in engineering problems using the concept of Laplace Transform. 2. Analyze engineering problems through the application of inverse Laplace transform of various functions. 3. Expand the periodic function by using the Fourier series for real-life problems and complex engineering problems. 4. Solve the problems of obtaining orthogonal trajectories and analytic functions by means of complex variable theory and application of harmonic conjugate. 5. Apply the concept of Correlation and Regression to the engineering problems in data science, machine learning, and AI. 6. Analyze the spread of data and distribution of probabilities by the concepts of probability and expectation.

Module No. & Name	Sub Topics	CO Mapped	Hrs/ Sub topic	Total Hrs/ Module
i. Prerequisites and Course Outline	Prerequisite Concepts and Course Introduction	**-**	**02**	**02**
1. Laplace Transform	Definition of Laplace transform, Condition of Existence of Laplace transform.	**CO1**	**01**	**07**
	Laplace Transform (L) of Standard Functions like eat, $sin(at)$, $cos(at)$, $sinh(at)$, $cosh(at)$ and tn, $n \geq 0$.		**02**	
	Properties of Laplace Transform: Linearity, First Shifting theorem, Second Shifting Theorem, change of scale Property, multiplication by t, Division by t, Laplace Transform of derivatives and integrals (Properties without proof).		**02**	
	Evaluation of integrals by using Laplace Transformation.		**02**	
2.Inverse Laplace Transform	Definition of Inverse Laplace Transform, Linearity property, Inverse Laplace Transform of standard functions, Inverse Laplace transform using derivatives.	**CO2**	**02**	**06**
	Partial fractions method to find inverse Laplace transform.		**02**	

	Inverse Laplace transform using Convolution theorem (without proof).		**02**	
3.Fourier Series	Dirichlet's conditions, Definition of Fourier series and Parseval's Identity (without proof).	**CO3**	**01**	**07**
	Fourier series of periodic function with period 2π and $2l$.		**02**	
	Fourier series of even and odd functions.		**02**	
	Fourier Transform-Fourier sine transform and Fourier cosine transform.		**02**	
4.Complex Variables	Function f(z) of complex variable, Limit, Continuity and Differentiability of f(z), Analytic function: Necessary and sufficient conditions for f(z) to be analytic (without proof).	**CO4**	**01**	**07**
	Cauchy-Riemann equations in Cartesian coordinates (without proof).		**02**	
	Milne-Thomson method to determine analytic function f(z) when real part (u) or Imaginary part (v) or its combination (u+v or u-v) is given.		**02**	
	Harmonic function, Harmonic conjugate and orthogonal trajectories.		**02**	
5.Statistical Techniques	Karl Pearson's coefficient of correlation (r)	**CO5**	**01**	**06**
	Spearman's Rank correlation coefficient (R) (with repeated and non-repeated ranks)		**01**	
	Lines of regression		**02**	
	Fitting of first- and second-degree curves.		**02**	
6.Probability	Definition and basics of probability, conditional probability.	**CO6**	**01**	**06**
	Total Probability theorem and Bayes' theorem.		**01**	
	Discrete and continuous random variable with probability distribution and probability density function.		**02**	
	Expectation, Variance, Moment generating function, Raw and central moments up to 4th order.		**02**	
ii. **Course Conclusion**	Recap of Modules, Outcomes, Applications, and Summarization.	**-**	**01**	**01**
Total Hours				**42**

Text Books:	1. Higher Engineering Mathematics, Dr. B. S. Grewal, Khanna Publication. 2. Advanced Engineering Mathematics, Erwin Kreyszig, Wiley Eastern Limited . 3. Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill Education.
Reference Books:	1. Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa publication. 2. Complex Variables and Applications, Brown and Churchill, McGraw-Hill Education. 3. Theory and Problems of Fourier Analysis with applications to BVP, Murray Spiegel, Schaum's Outline Series.
Useful Links:	1. e-PGPathshala (inflibnet.ac.in) 2. https://nptel.ac.in/noc/courses/111/ 3. https://www.coursera.org/courses?query=mathematics 4. https://ndl.iitkgp.ac.in/

Term Work (TW)	1. Each Student has to write at least 6 class tutorials on entire syllabus. 2. Journal must include at least 2 assignments on content of theory of the course. The distribution of Term Work marks will be as follows – • Class Tutorials on entire syllabus:15 marks • Assignment: 10 marks
Assessment:	
Continuous Assessment for 40 marks: 1. Test 1 – 30 marks 2. Test 2 – 30 marks 3. Internal assessment - 10 marks Internal assessment will be based on assignments/quizzes /case study/activity conducted by the faculty	
End Semester Theory Examination will be of 60-Marks for 02 hrs 30 min duration.	

Course Code	Course Name	Credits (TH+P+TUT)
CEC302	Discrete Structures and Graph Theory	2-0-0

Prerequisite:	Basic Mathematics
Course Objectives:	1. Cultivate clear thinking and creative problem solving. 2. Thoroughly train in the construction and understanding of mathematical proofs. 3. Exercise common mathematical arguments and proof strategies. 4. To apply graph theory in solving practical problems. 5. Thoroughly prepare for the mathematical aspects of other Computer Engineering courses
Course Outcomes:	**After the successful completion of this course, learner will be able to:** 1. Have an ability to reason logically. 2. Solve problems on relations and functions techniques. 3. Emphasize the concept of Posets and Lattice 4. Use counting techniques to representation and characterization of relational concept. 5. Use groups and codes in Encoding-Decoding 6. Apply concepts of graph theory in solving real world problems.

Module No. & Name	Sub Topics	CO mapped	Hrs / Sub topics	Total Hrs / Module
i. Prerequisites and Course outline	Prerequisite Concepts and Course Introduction.	**-**	**02**	**02**
1. Logic	Propositional Logic, Predicate Logic, Laws of Logic, Quantifiers, Normal Forms	**CO1**	**02**	**04**
	Inference Theory of Predicate Calculus, Mathematical Induction		**02**	
2. Relations and Functions	Basic concepts of Set Theory	**CO2**	**01**	**04**
	Relations: Definition, Types of Relations, Representation of Relations, Closures of Relations, Warshall's algorithm, Equivalence relations and Equivalence Classes		**02**	
	Functions: Definition, Types of functions, Com position of functions, Identity and Inverse function		**01**	
3. Posets and Lattice	Partial Order Relations, Poset, Hasse Diagram	**CO3**	**02**	**04**
	Chain and Anti chains, Lattice, Types of Lattices, Sub lattice		**02**	
4. Counting	Basic Counting Principle-Sum Rule, Product Rule, Inclusion- Exclusion Principle, Pigeonhole Principle	**CO4**	**02**	**04**
	Recurrence relations, Solving recurrence relations		**02**	
5. Algebraic Structures	**Algebraic structures with one binary operation:** Semi group, Monoid, Groups, Subgroups, Abelian Group, Cyclic group, Isomorphism	**CO5**	**02**	**05**
	Algebraic structures with two binary operations: Ring		**02**	

	Coding Theory: Coding, binary information and error detection, decoding and error correction		**03**	
6. Graphs	Types of graphs, Graph Representation, Sub graphs, Operations on Graphs, Walk, Path, Circuit, Connected Graphs, Disconnected Graph, Components	**CO6**	**03**	**05**
	Homomorphism and Isomorphism of Graphs, Euler and Hamiltonian Graphs, Planar Graph, Cut Set, Cut Vertex, Applications		**02**	
ii. Course conclusion	Recap of Modules, Outcomes, Applications, and Summarization.	**--**	**01**	**01**
Total Hrs				**28**

Books:	
Textbooks	1. Bernad Kolman, Robert Busby, Sharon Cutler Ross, Nadeem-ur-Rehman, "Discrete Mathematical Structures", Pearson Education. 2. C. L. Liu "Elements of Discrete Mathematics", second edition 1985, McGraw-Hill Book Company. Reprinted 2000. 3. K. H. Rosen, "Discrete Mathematics and applications", fifth edition 2003, Tata McGraw Hill Publishing Company
Reference Books	1. Y N Singh, "Discrete Mathematical Structures", Wiley-India. 2. J. L. Mott, A. Kandel, T. P. Baker, "Discrete Mathematics for Computer Scientists and Mathematicians", Second Edition 1986, Prentice Hall of India. 3. J. P. Trembley, R. Manohar "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill Publishing Company 4. Seymour Lipschutz, Marc Lars Lipson, "Discrete Mathematics" Schaum"s Outline, McGraw Hill Education. 5. Narsing Deo, "Graph Theory with applications to engineering and computer science", PHI Publications. 6. P. K. Bisht, H. S. Dhami, "Discrete Mathematics", Oxford press.

Useful Links:

1. https://www.edx.org/learn/discrete-mathematics
2. https://www.coursera.org/specializations/discrete-mathematics
3. https://nptel.ac.in/courses/106/106/106106094/
4. https://swayam.gov.in/nd1_noc19_cs67/preview

Assessment:

Continuous Assessment for 40 marks:

4. Test 1 – 20 marks
5. Test 2 – 20 marks
6. Internal assessment - 10 marks

Internal assessment will be based on assignments/quizzes /case study/activity conducted by the faculty

End Semester Theory Examination will be of 45 marks of 2 hrs duration.

Course Code	Course Name	Credits (TH+P+TUT)
CEC303	**Data Structure**	3 - 0 – 0

Prerequisite:	C programming
Course Objectives:	1. To discuss types of different data structures and concept of Abstract Data Type 2. To discuss the concept of stack and queue and apply them to various applications. 3. To describe the concept of link list and apply it to various applications 4. To introduce the different kinds of trees. 5. To discuss graph related concepts and traversals along with application. 6. To teach various searching techniques.
Course Outcomes:	**After successful completion of this course, learner will be able to:** 1. Describe types of data structure and write ADT. 2. Implement stack and different types of queues using array and their applications 3. Perform various types of link list operations and their applications 4. Perform operations on Binary Search Tree, AVL tree, Btree and B+Tree 5. Implement Graph traversals BFS, DFS and application of Graph in topological sorting 6. Describe various Hashing functions, Collision techniques and compare various searching techniques Linear Search, Binary Search and Hashing

Module No. & Name	Sub Topics	CO mapped	Hrs / Sub Topics	Total Hrs/ Module
i. Prerequisites and Course outline	Prerequisite Concepts and Course Introduction.	**-**	**02**	**02**
1. Introduction to Data Structures	Introduction to Data Structures, Concept of ADT,	**CO1**	**01**	**02**
	Types of Data Structures-Linear and Nonlinear, Operations on Data Structures.		**01**	
2.Stack and Queues	Introduction, ADT of Stack, Operations on Stack, Array Implementation of Stack	**CO2**	**01**	**09**
	Applications of Stack-Well formedness of Parenthesis		**01**	
	Infix to Postfix Conversion		**01**	
	Postfix Evaluation		**01**	
	Recursion		**01**	
	Introduction, ADT of Queue, Operations on Queue, Array Implementation of Queue		**01**	
	Implementation of circular and Double Ended Queue, Priority Queue, Applications of Queue		**03**	
3. Linked List	Introduction, Representation of Linked List, Linked List v/s Array	**CO3**	**01**	**10**

	Types of Linked List - Singly Linked List, Circular Linked List, Doubly Linked List, Operations on Singly Linked List and Doubly Linked List		**06**	
	Stack and Queue using Singly Linked List		**01**	
	Singly Linked List Application-Polynomial Representation and Addition		**02**	
4. Trees	Introduction, Tree Terminologies, Binary Tree, Binary Tree Representation,Types of Binary Tree	**CO4**	**01**	**11**
	Binary Tree Traversals		**02**	
	Binary Search Tree, Operations on Binary Search Tree		**04**	
	Applications of Binary Tree-Expression Tree, Huffman Encoding		**01**	
	Search Trees-AVL, rotations in AVL Tree, operations on AVL Tree, Introduction of B Tree, B+ Tree		**03**	
5. Graphs	Introduction, Graph Terminologies, Representation of Graph	**CO5**	**01**	**04**
	Graph Traversals-Depth First Search (DFS) and Breadth First Search (BFS)		**02**	
	Graph Application- Topological Sorting		**01**	
6. Searching Techniques	Linear Search, Binary Search, Hashing-Concept, Hash Functions	**CO6**	**01**	**03**
	CollisionResolution Techniques		**02**	
ii. Course conclusion	Recap of Modules, Outcomes, Applications, and Summarization.	**--**		**01**
Total Hours				**42**

Books:

Textbooks:

1. Aaron M Tenenbaum, Yedidyah Langsam, Moshe J Augenstein, “Data Structures Using C”, Pearson Publication.
2. Reema Thareja, “Data Structures using C”, Oxford Press.
3. Richard F. Gilberg and Behrouz A. Forouzan, “Data Structures: A Pseudocode Approach with C”, 2ndEdition, CENGAGE Learning.
4. Jean Paul Tremblay, P. G. Sorenson, “Introduction to Data Structure and Its Applications”, McGraw-Hill Higher Education
5. Data Structures Using C, ISRD Group, 2ndEdition, Tata McGraw-Hill.

Reference Books:

1. Prof. P. S. Deshpande, Prof. O. G. Kakde, “C and Data Structures”, DreamTech press.
2. E. Balagurusamy, “Data Structure Using C”, Tata McGraw-Hill Education India.
3. Rajesh K Shukla, “Data Structures using C and C++”, Wiley-India
4. GAV PAI, “Data Structures”, Schaum’s Outlines.
5. Robert Kruse, C. L. Tondo, Bruce Leung, “Data Structures and Program

	Design in C", Pearson Edition
Useful Links:	
1. https://nptel.ac.in/courses/106/102/106102064/ 2. https://www.coursera.org/specializations/data-structures-algorithms 3. https://www.edx.org/course/data-structures-fundamentals 4. https://swayam.gov.in/nd1_noc19_cs67/preview	
Assessment:	
Continuous Assessment for 40 marks: 1. Test 1 – 30 marks 2. Test 2 – 30 marks 3. Internal assessment - 10 marks Internal assessment will be based on assignments/quizzes /case study/activity conducted by the faculty	
End Semester Theory Examination will be of 60 marks of 02 hrs min 30 duration.	

Course Code	Course Name	Credits (TH+P+TUT)
CEC304	Digital Logic & Computer Architecture	3-0-0
Prerequisite:	Knowledge on number systems	
Course Objectives:	1. To have the rough understanding of the basic structure and operation of basic digital circuits and a digital computer. 2. To discuss in detail arithmetic operations in digital systems. 3. To discuss generation of control signals and different ways of communication with I/O devices. 4. To study the hierarchical memory and principles of advanced computing.	
Course Outcomes:	**After the successful completion of this course, learner will be able to:** 1. Learn different number systems and basic structure of computer systems. 2. Demonstrate the arithmetic algorithms. 3. Describe the basic concepts of digital components and processor organization. 4. Explain the generation of control signals of computers. 5. Demonstrate the memory organization. 6. Describe the concepts of parallel processing and different Buses.	

Module No. & Name	Sub Topics	CO mapped	Hrs / Sub Topics	Total Hrs/ Module
i. Prerequisites and Course outline	Prerequisite Concepts and Course Introduction.	**-**	**02**	**02**
1. Computer Fundamentals	Introduction to Number System and Codes Number Systems: Binary, Octal, Decimal, Hexadecimal	**CO1**	**01**	**05**
	Codes: Grey, BCD, Excess-3, ASCII, Boolean Algebra		**02**	
	Logic Gates: AND, OR, NOT, NAND, NOR, EX-OR		**01**	
	Overview of computer organization and architecture. Basic Organization of Computer and Block Level functional Units, Von- Neumann Model		**01**	
2. Data Representation and Arithmetic algorithms	Binary Arithmetic: Addition, Subtraction, Multiplication	**CO1, CO2**	**01**	**08**
	Division using Sign Magnitude, 1's and 2's compliment		**02**	
	BCD and Hex Arithmetic Operation		**01**	
	Booths Multiplication Algorithm, Restoring and Non-restoring Division Algorithm. IEEE-754 Floating point Representation		**04**	

3.Processor Organization and Architecture	Introduction: Half adder, Full adder, MUX, DMUX, Encoder, Decoder(IC level)	**CO3**	**02**	**06**
	Introduction to Flip Flop: SR, JK, D, T (Truth table)		**02**	
	Register Organization, Instruction Formats, Addressing modes, Instruction Cycle, Interpretation and sequencing		**02**	
4. Control Unit Design	Hardwired Control Unit: State Table Method, Delay Element Methods	**CO4**	**03**	**06**
	Microprogrammed Control Unit: Micro Instruction-Format, Sequencing and execution, Micro operations, Examples of microprograms		**03**	
5. Memory Organization	Introduction and characteristics of memory, Types of RAM and ROM, Memory Hierarchy, 2-level Memory Characteristic	**CO5**	**03**	**06**
	Cache Memory: Concept, locality of reference, Design problems based on mapping techniques, Cache coherence and write policies. Interleaved and Associative Memory		**03**	
6. Principles of Advanced Processor and Buses	Basic Pipelined Data path and control, data dependencies	**CO6**	**02**	**08**
	Data hazards, branch hazards, delayed branch, and branch prediction, Performance measures-CPI, Speedup, Efficiency, throughput		**02**	
	Amdhal's law. Flynn's Classification, Introduction to multicore architecture		**02**	
	Introduction to buses: ISA, PCI, USB. Bus Contention and Arbitration		**02**	
ii.Course conclusion	Recap of Modules, Outcomes, Applications, and Summarization.	**--**	**01**	**01**
Total Hours				**42**

Books:	
Textbooks	1. R. P. Jain, "Modern Digital Electronic", McGraw-Hill Publication, 4thEdition. 2. William Stalling, "Computer Organization and Architecture: Designing and Performance", Pearson Publication 10TH Edition. 3. John P Hayes, "Computer Architecture and Organization", McGraw-Hill Publication, 3RD Edition. 4. Dr. M. Usha and T. S. Shrikanth, "Computer system Architecture and Organization", Wiley publication.
Reference Books	1. Andrew S. Tanenbaum, "Structured Computer Organization", Pearson Publication. 2. B. Govindarajalu, "Computer Architecture and Organization", McGraw-Hill Publication. 3. Malvino, "Digital computer Electronics", McGraw-Hill Publication, 3rd edition.

	4. Smruti Ranjan Sarangi, "Computer Organization and Architecture", McGraw-Hill Publication.

Useful Links:

1. https://www.classcentral.com/course/swayam-computer-organization-and-architecture-a-pedagogical- aspect-9824
2. https://nptel.ac.in/courses/106/103/106103068/
3. https://www.coursera.org/learn/comparch
4. https://www.edx.org/learn/computer-architecture

Assessment:

Continuous Assessment for 40 marks:

1. Test 1 – 30 marks
2. Test 2 – 30 marks
3. Internal assessment --10 marks

Internal assessment will be based on assignments/quizzes /case study/activity conducted by the faculty

End Semester Theory Examination will be of 60 marks of 02 hrs 30 min duration.

Course Code	Course Name	Credits (TH+P+TUT)
CEC305	**Computer Graphics**	3 - 0 - 0

Prerequisite:	Knowledge of C Programming and Basic Mathematics**.**
Course Objectives:	1. To equip students with the fundamental knowledge and basic technical competence in the field of Computer Graphics. 2. To emphasize on implementation aspect of Computer Graphics Algorithms. 3. To prepare the student for advance areas and professional avenues in the field of Computer Graphics
Course Outcomes:	**At the end of the course, students should be able to** 1. Describe the basic concepts of Computer Graphics 2. Demonstrate various algorithms for basic graphics primitives 3. Apply 2-D geometric transformations on graphical objects 4. Use various Clipping algorithms on graphical objects 5. Apply 3-D geometric transformations, curve representation techniques and projections methods 6. Explain visible surface detection techniques and Animation.

Module No. & Name	Sub Topics	CO mapped	Hrs / Sub topics	Total Hrs / Module
i. Prerequisites and Course outline	Prerequisite Concepts and Course Introduction.	**-**	**02**	**02**
1. Introduction and Overview of Graphics System	Definition and Representative uses of computer graphics, Overview of coordinate system, Definition of scan conversion, Rasterization and Rendering	**CO1**	**01**	**03**
	Raster scan & Random scan displays, Architecture of Raster graphics system with display processor, Architecture of Random scan systems. Self-learning topics: Display devices like Plasma Display, 3D Display		**02**	
2. Output Primitives	Scan conversions of point, line, circle and ellipse: DDA algorithm and Bresenham algorithm for line drawing, midpoint algorithm for circle, midpoint algorithm for ellipse drawing (Mathematical derivation for above algorithms is expected)	**CO2**	**08**	**12**
	Aliasing, Antialiasing techniques like Pre and post filtering, super sampling, and pixel phasing)		**01**	

	Filled Area Primitive: Scan line Polygon Fill algorithm, inside outside tests, Boundary Fill and Flood fill algorithm		**03**	
3. Two Dimensional Geometric Transformations	3D Object representation methods, Basic transformations: Translation, Scaling, Rotation	**CO3**	**02**	**04**
	Matrix representation and Homogeneous Coordinates		**01**	
	Composite transformation, Other transformations: Reflection and Shear		**01**	
4. Two-Dimensional Viewing and Clipping	Viewing transformation pipeline and Window to Viewport coordinate transformation	**CO4**	**02**	**06**
	Clipping operations: Point clipping, Line clipping algorithms: Cohen-Sutherland, Liang: Barsky, Polygon Clipping Algorithms: Sutherland- Hodgeman, Weiler-Atherton		**04**	
5. Three Dimensional Geometric Transformations, Curves and Fractal Generation	3D Transformations: Translation, Rotation, Scaling and Reflection.	**CO5**	**01**	**08**
	Composite transformations: Rotation about an arbitrary axis		**01**	
	Projections – Parallel, Perspective. (Matrix Representation)		**02**	
	Bezier Curve, B-Spline Curve, Fractal-Geometry: Fractal Dimension, Koch Curve. **Self-learning topics**: Piano Curve, Hilbert Curve		**04**	
6. Visible Surface Detection and Animation	Visible Surface Detection: Classification of Visible Surface Detection algorithm, Back Surface detection method, Depth Buffer method, Area Subdivision method	**CO6**	**03**	**06**
	Animation: Introduction to Animation, Traditional Animation Techniques, Principles of Animation, Key framing: Character and Facial Animation, Deformation, Motion capture		**03**	
i. Course conclusion	Recap of Modules, Outcomes, Applications, and Summarization.	**--**	**01**	**01**
Total Hours				**42**
Books:				
Textbooks	1. Hearn & Baker, "Computer Graphics C version", 2nd Edition, Pearson Publication 2. James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice in C", 2ndEdition, Pearson Publication 3. Samit Bhattacharya, "Computer Graphics", Oxford publication			
Reference Books	1. D. Rogers, "Procedural Elements for Computer Graphics", Tata McGraw-Hill Publications. 2. Zhigang Xiang, Roy Plastock, "Computer Graphics", Schaum"s Outlines McGraw-Hill Education 3. Rajesh K. Maurya, "Computer Graphics", Wiley India Publication.			

	4. F. S. Hill, “Computer Graphics using OpenGL”, Third edition, Pearson Publications.

Useful Links:

1. https://onlinecourses.nptel.ac.in/noc22_cs111/preview
2. https://nptel.ac.in/courses/106/106/106106090/
3. https://www.classcentral.com/course/interactivegraphics-2067

Assessment:

Continuous Assessment for 40 marks:

1. Test 1 – 30 marks
2. Test 2 – 30 marks
3. Internal assessment - 10 marks

Internal assessment will be based on assignments/quizzes /case study/activity conducted by the faculty

End Semester Theory Examination will be of 60 marks for 2hr 30min duration.

Lab Code	Lab Name	Credits (P+TUT)
CEL303	**Data Structures Lab**	1- 0

Lab Prerequisite:	C Programming
Lab Objectives:	1. To implement basic data structures such as linked lists, stacks and queues 2. To solve problem involving graphs and trees 3. To choose appropriate data structure and apply it to various problems
Lab Outcomes (LOs):	**At the end of the course, the student will be able to** 1. Implement linear data structures & be able to handle operations like insertion, deletion, searching and traversing on them. 2. Implement nonlinear data structures & be able to handle operations like insertion, deletion, searching and traversing on them 3. Choose appropriate data structure and apply it in various problems 4. Select appropriate searching techniques for given problems. 5. Apply ethical principles like timeliness and adhere to the rules of the laboratory.

Lab No.	Experiment Title	LO mapped	Hrs / Lab
0	Prerequisite	**-**	**02**
1	Implement Stack ADT using array.	**LO1, LO5**	**02**
2	Convert an Infix expression to Postfix expression using stack ADT.	**LO1, LO3, LO5**	**02**
3	Evaluate Postfix Expression using Stack ADT.	**LO1, LO3, LO5**	**02**
4*	At least 2 applications of Stack from the **useful links/any other** given below.	**LO1, LO3, LO5**	**02**
5	Implement Linear Queue ADT using array.	**LO1, LO3, LO5**	**02**
6	Implement Circular/Double ended Queue ADT using array.	**LO1, LO3, LO5**	**02**
7	Implement Priority Queue ADT using array.	**LO1, LO3, LO5**	**02**
8	Implement Singly Linked List ADT.	**LO1, LO3, LO5**	**02**
9	Implement Circular Linked List ADT.	**LO1, LO3, LO5**	**02**
10	Implement Doubly Linked List ADT.	**LO1, LO3, LO5**	**02**
11	Implement Stack / Linear Queue ADT using Linked List.	**LO1, LO3, LO5**	**02**
12*	Implement Binary Search Tree ADT using Linked List.	**LO2, LO3, LO5**	**02**
13*	Implement Graph Traversal techniques:) Depth First Search b) Breadth First Search	**LO2, LO3, LO5**	**02**
14*	At least 2 applications of Binary Search Technique from the **useful links/any other** given below	**LO4, LO5**	**02**

Useful Links:

1. www.leetcode.com

2. www.hackerrank.com
3. www.cs.usfca.edu/~galles/visualization/Algorithms.html
4. www.codechef.com
5. https://learndsa.kjsieit.in/

Term work:

1. Term work should consist of 10 experiments.
2. **star (*) marked experiments are compulsory**.
3. Journal must include at least 2 assignments.
4. The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
5. Total 25 Marks (Experiments: 20-marks, Assignments: 05-marks)

Oral & Practical Exam:

Oral & Practical Exam will be based on the entire syllabus of CEC303 and CEL303

Lab Code	Lab Name	Credit(P+TUT)
CEL304	**Digital Logic & Computer Architecture Lab**	**1-0**

Lab Prerequisite:	C Programming Language
Lab Objectives:	1. To implement operations of the arithmetic unit using algorithms. 2. Design and simulate different digital circuits. 3. To design memory subsystems including cache memory. 4. To demonstrate CPU and ALU design.
Lab Outcomes (LOs):	**At the end of the course, the student will be able to** 1. Describe the basics of digital components 2. Design the basic building blocks of a computer: ALU, registers, CPU and memory 3. Recognize the importance of digital systems in computer architecture 4. Implement various algorithms for arithmetic operations. 5. Apply ethical principles like timeliness and adhere to the rules of the laboratory.

Lab No.	Experiment Title	LO mapped	Hrs/Lab
0	Prerequisite	**-**	**02**
1	To verify the truth table of various logic gates using ICs.	**LO1, LO5**	**02**
2	To realize the gates using universal gates	**LO1, LO5**	**02**
3	Code conversion.	**LO1, LO5**	**02**
4	To realize half adder and full adder.	**LO2, LO5**	**02**
5	To implement logic operation using MUX IC.	**LO3, LO5**	**02**
6	To implement logic operation decoder IC.	**LO3, LO5**	**02**
7	Study of flip flop IC.	**LO3, LO5**	**02**
8	To implement ripple carry adder.	**LO3, LO5**	**02**
9	To implement carry look ahead adder.	**LO3, LO5**	**02**
10	To implement Booth's algorithm.	**LO4, LO5**	**02**
11	To implement a restoring division algorithm.	**LO4, LO5**	**02**
12	To implement non restoring division algorithm.	**LO4, LO5**	**02**
13	To implement ALU design.	**LO2, LO5**	**02**
14	To implement CPU design.	**LO2, LO5**	**02**
15	To implement memory design.	**LO2, LO5**	**02**
16	To implement cache memory design.	**LO2, LO5**	**02**

Note:

1. Any Four experiments from Exp. No. 1 to Exp. No. 7 using hardware.
2. Any Six experiments from Exp. No. 8 to Exp. No. 16 using Virtual Lab, expect Exp. No. 10,11 and 12.
3. Exp. No. 10 to Exp. No. 12 using Programming language.

Useful Link:

Link http://cse10-iitkgp.virtual-labs.ac.in/

Term work:

1. Term work should consist of minimum 10 experiments

2. Journal must include at least 2 assignments on content of theory and practical of the course “Digital Logic &Computer Organization and Architecture”
3. The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4. Total 25 Marks (Experiments: 20-marks, Assignments: 05-marks)

Lab Code	Lab Name	Credits (P)
CEL305	**Computer Graphics Lab**	**01**
Lab Prerequisite:	C Programming Language	
Lab Objectives:	1. Understand the need of developing graphics application 2. Learn algorithmic development of graphics primitives like line, circle, polygon etc. 3. Learn the representation and transformation of graphical images and pictures	
Lab Outcomes (LOs):	**At the end of the lab, students will be able to:** 1. Implement various output and filled area primitive algorithms 2. Apply transformation, projection and clipping algorithms on graphical objects 3. Perform curve and fractal generation methods 4. Develop a Graphical application/Animation based on learned concept 5. Apply ethical principles like timeliness and adhere to the rules of the laboratory.	

Content:

Scan conversions: lines, circles, ellipses. Filling algorithms, clipping algorithms. 2D and 3D transformation Curves Visible surface determination. Simple animations Application of these through exercises in C/C++/OpenGL

Lab No.	Experiment Title	LO mapped	Hrs/ Lab
0	Prerequisite	**-**	**02**
1	Implement DDA Line Drawing algorithm (dotted/dashed/thick)	**LO1, LO5**	**02**
2	Implement Bresenham's Line algorithm(dotted/dashed/thick)	**LO1, LO5**	**02**
3	Implement midpoint Circle algorithm.	**LO1, LO5**	**02**
4	Implement midpoint Ellipse algorithm.	**LO1, LO5**	**02**
5	Implement Area Filling Algorithm: Boundary Fill, Flood Fill.	**LO1, LO5**	**02**
6	Implement Scan line Polygon Filling algorithm.	**LO1, LO5**	**02**
7	Implement Curve: Bezier for n control points, B Spline (Uniform)(at least one)	**LO3, LO5**	**02**
8	Implement Fractal generation method (anyone)	**LO3, LO5**	**02**
9	Character Generation: Bit Map method and Stroke Method	**LO1, LO5**	**02**
10	Implement 2D Transformations: Translation, Scaling, Rotation, Reflection, Shear.	**LO2, LO5**	**02**
11	Implement Line Clipping Algorithm: Cohen Sutherland / Liang Barsky.	**LO2, LO5**	**02**
12	Implement polygon clipping algorithm (at least one)	**LO2, LO5**	**02**
13	Program to perform 3D transformation.	**LO2, LO5**	**02**
14	Perform projection of a 3D object on Projection Plane: Parallel and Perspective.	**LO2, LO5**	**02**
15	Perform Animation (such as Rising Sun, Moving Vehicle, Smileys, Screen saver etc. using C/C++/Java/OpenGL/Blender/ any other tool)	**LO1, LO2, LO3, LO4, LO5**	**02**

16.	Case Study: Virtual Reality and Sample program using VRML	**LO4, LO5**	**02**

Virtual Lab Links:

http://vlabs.iitb.ac.in/vlabs-dev/labs/cglab/experimentlist.html

Term work:

1. Term work should consist of 10 experiments.
2. Journal must include at least 2 assignments
3. The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4. Total 25 Marks (Experiments: 20-marks, Assignments: 05-marks)

Oral & Practical exam:

Oral & Practical Exam will be based on the entire syllabus of **CEC305** and **CEL305**

Course code	Course Name	Credits
CEPR31	**Project Based Learning: Mini Project Lab-I**	**01**

PBL Objectives:	1. To acquaint with the process of identifying the needs and converting it into the problem. 2. To familiarize the process of solving the problem in a group. 3. To acquaint with the process of applying basic engineering fundamentals to attempt solutions to the problems. 4. To inculcate the process of self-learning and research.
PBL Outcomes (PROs):	**At the end of the course, the student will be able to:** 1. Identify problems based on societal /research needs. 2. Apply Knowledge and skill to solve societal problems in a group. 3. Develop interpersonal skills to work as member of a group or leader. 4. Analyze the impact of solutions in societal and environmental context for sustainable development. 5. Excel in written and oral communication. 6. Demonstrate capabilities of self-learning in a group, which leads to lifelong learning. 7. Demonstrate project management principles during project work.

Guidelines for Mini Project

1	Project based learning Mini Project Lab-1 should be implemented preferably using Java programming (CEXS33)
2	Students shall form a group of 2 to 3 students, while forming a group shall not be allowed less than two or more than three students, as it is a group activity.
3	Students should do survey and identify needs, which shall be converted into problem statement for mini project in consultation with faculty supervisor/internal committee of faculties.
4	Students shall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of mini project.
5	A logbook to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.
6	Faculty supervisor may give inputs to students during mini project activity; however, focus shall be on self-learning.
7	Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.
8	Students shall convert the best solution into working model using Java programming.
9	The solution to be validated with proper justification and report to be compiled in standard format of the college.

10	With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the Mini Projects, it is preferable that a single project of appropriate level and quality to be carried out in two semesters by all the groups of the students. i.e. Mini Project 1 in semester III and IV.
11	However, based on the individual students or group capability, with the mentor's recommendations, if the proposed Mini Project adhering to the qualitative aspects mentioned above gets completed in odd semester, then that group can be allowed to work on the extension of the Mini Project with suitable improvements/modifications or a completely new project idea in even semester. This policy can be adopted on case by case basis.

Term Work:

The review/ progress monitoring committee shall be constituted by senior faculty members. The progress of mini project to be evaluated on continuous basis, minimum two reviews in each semester. Assessment also considers peer review and ethics observed by faculties and participation involvement.

Continuous Assessment:

In continuous assessment focus shall also be on each individual student, log book maintained and weekly meeting based on the same..

Distribution of Term work marks for both semesters shall be as below:	**Practical Marks**
Marks awarded by guide/supervisor based on implementation	10
Peer assessment by team members	5
Marks awarded by review committee	5
Quality of Project report	5

Review / progress monitoring committee may consider following points for assessment based on project as mentioned in general guidelines

1	Students' group shall complete project in all aspects including, a. Identification of need/problem b. Proposed final solution c. Procurement of components/system d. Building prototype and testing
2	Continuous assessment will be weekly based on logbook. Two presentations will be conducted for review before a panel. a. First shall be for finalization of problem and proposed solution

	b. Second shall be for implementation and testing of solution.

Assessment criteria of Mini Project

Mini Project shall be assessed based on following criteria:

1	Quality of survey and identification of problem statement
2	Innovativeness in solutions
3	Implementation
4	Team work
5	Project report

Guidelines for Assessment of Mini Project Practical/Oral Examination:

Report should be prepared as per the guidelines issued by the college.
Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of examiners.
Students shall be motivated to participate in poster & project competition on the work in students' competitions.

Mini Project shall be assessed based on following points;

1. Quality of problem and Clarity
2. Innovativeness in solutions
3. Cost effectiveness and Societal impact
4. Full functioning of working model as per stated requirements
5. Effective use of skill sets
6. Effective use of standard engineering norms
7. Contribution of an individuals as member or leader
8. Clarity in written and oral communication

Assessment:

Term Work for 25 Marks:

Term work will be based on assessment of Project Implementation and a Logbook which is filled by students on weekly basis as per their weekly progress.

Oral and Practical Exam for 25 Marks:

Based on Project Implementation

Course Code	Course Name	Credits
CEXS33	Skill Based Learning: Object Oriented Programming with Java (SAT-III)	01

Prerequisite:	Structured Programming Approach
Skill Objectives:	1. To learn the basic concepts of object-oriented programming 2. To study JAVA programming language 3. To study various concepts of JAVA programming like multithreading, exception Handling, packages, etc. 4. To explain components of GUI based programming.
Skill Outcomes (SOs):	**At the end of the course, the student will be able to** 1. Apply fundamental programming constructs. 2. Implement the concept of classes and objects, inheritance and interfaces. 3. Implement the concept of strings, arrays, vectors and packages 4. Implement the concept of exception handling and multithreading. 5. Develop GUI based application. 6. Apply ethical principles like timeliness and adhere to the rules of the laboratory

Lab No.	Experiment Title	SO mapped	Hrs /Lab
1	**Title:** Write a program to implement basic programming constructs like branching and looping. **Concepts:** Introduction to Java, Object Oriented Concepts, Java Virtual Machine, Basic programming constructs: variables, data types, and operators, expressions, branching and looping.	**SO1, SO6**	**02**
2	Write a program to demonstrate different ways of accepting user input in Java. **Concepts:** Class, object, data members, member functions, Command Line Argument, Input and output functions in Java, Buffered reader class, Scanner class.	**SO1, SO6**	**02**
3	Write a program to implement the concept of 1. Method overloading 2. Constructor overloading. **Concepts:** Method, how to pass parameters, Method overloading, static members and functions, Introduction to Constructors, Constructor types, Constructor overloading.	**SO2, SO6**	**02**
4	Write a program implement the concept of 2D array and String Manipulation functions in Java. **Concepts:** Array, Strings, String Buffer	**SO3, SO6**	**02**
5	Write a program to implement the concept of Inheritance. **Concepts:** Inheritance, Types of inheritance, extends keyword , super keyword, Access Modifiers	**SO2, SO6**	**02**

6	Write a program to implement the concept of Method Overriding. **Concepts:** Inheritance, Method Overriding.	**SO2, SO6**	**02**
7	Write a program to implement the concept of abstract class and abstract method. **Concepts:** Abstract class and abstract method	**SO2, SO6**	**02**
8	Write a program to implement the concept of package. **Concepts:** Introduction to Packages, Types of Packages-Built-in packages, User defined packages	**SO3, SO6**	**02**
9	Write a program to implement the concept of Exception handling **Concepts:** Exception handling using try, catch, finally, throw and throws, Multiple try and catch blocks, User Defined Exceptions	**SO4, SO6**	**02**
10	Write a program to implement the concept of Multithreading **Concepts:** Introduction to Multithreading, Thread lifecycle, thread class methods, creating threads using extends and implements keyword.	**SO4, SO6**	**02**
11	Design form for Admission process management application system using AWT or Java Swing **Concepts:** Applet and applet life cycle, creating applets, graphics class functions, parameter passing to applet, Font and color class. Event handling using event class AWT: working with windows, using AWT controls for GUI design Swing class in JAVA.	**SO5, SO6**	**02**
12	Study and Implement the concept of JDBC and Perform CRUD Operation on the form created in 11 using Java Database Connectivity **Concepts:** Introduction to JDBC, JDBC-ODBC connectivity, JDBC architecture.	**SO5, SO6**	**02**

Textbooks

1. Herbert Schildt, 'JAVA: The Complete Reference', Ninth Edition, Oracle Press.
2. E. Balagurusamy, 'Programming with Java', McGraw Hill Education.

Reference Books

1. "JAVA Programming", Black Book, Dreamtech Press
2. Dietaland Dietal, "Java: How to Program", 8th Edition, PHI
3. Ivor Horton, "Beginning JAVA", Wiley India
4. "Learn to Master Java programming", Staredu Solutions

Useful Links:

1. www.nptelvideos.in
2. www.w3schools.com
3. www.tutorialspoint.com
4. https://starcertification.org/Certifications/Certificate/securejava

Virtual Lab Link:

http://vlabs.iitb.ac.in/vlabs-dev/vlab_bootcamp/bootcamp/bots_with_dots/labs/index.html

Assessment:

Term Work for 25 Marks:

Programming labs to be conducted as 2hrs continuous theory + hands-on session. The assessment will be

- An online quiz conducted at the end of every 2-hr session consisting of 5 questions for a total of 10 marks. The average of best 10 quizzes will be considered toward 10 marks out of 25.

- Students should perform minimum 12 experiments. The programs performed along with the screenshot of output have to be submitted within two days. A cover page will be attached stating the aims and objectives. This will be considered towards 10 marks

	Quiz	**Lab Submission**	**Total**
Marks Allotted	10	15	25

Exposure Course Code	Exposure Course Name	Credits			
		TH	P	TUT	Total
CEXS34	SAT – IV: Activity-Based Learning (Interdisciplinary Informatics)	-	01	-	01
ABL Objectives (AOBs):	1. To expose learners to the opportunities, effectiveness and benefits of integrating informatics with diverse disciplines such as biotechnology, healthcare, agriculture, nanotechnology, earth sciences, etc. 2. To introduce the approaches for integrating informatics with different disciplines. 3. To explore real-world applications of interdisciplinary informatics, relevant data and tools for its development. 4. To acquaint learners with recent trends and research in interdisciplinary informatics. 5. To enhance critical thinking, research, communication and presentation skills. 6. To promote interdisciplinary research and development.				
ABL Outcomes (AOs):	**Upon completion of the course, the learners will be able to:** 1. Understand the fundamental concepts and interdisciplinary nature of informatics. 2. Analyze literature, case studies and successful solutions related to interdisciplinary informatics applications. 3. Analyze and interpret the data for interdisciplinary informatics. 4. Identify real-world problems that can be addressed through interdisciplinary informatics. 5. Demonstrate effective communication skills to bridge the gap between disciplinary jargons and develop interdisciplinary collaborations. 6. Demonstrate a life-long motivation to engage in hands-on projects, research and practices in sustainable interdisciplinary informatics.				
Guidelines for ActivityBased Learning (ABL):	1. Students shall work in team of 03-04 members, which shall remain for this entire course. 2. Student teams shall choose, survey and study any 01 of the following informatics using the Internet / Library Resources / Research Articles / Case Study Reports / etc.: i. Bioinformatics ii. Agro Informatics iii. Health Informatics iv. Weather Informatics v. Nano Informatics vi. Geo Informatics 3. Students are also required to study the recent Research and Development in the interdisciplinary informatics, focusing on need-based real-world applications. 4. During the contact hours, each student team is required to provide a weekly report of their progress — orally and as written summaries of approximately 01-02 pages, accompanied by a list of references. 5. During the contact hours across the entire semester, each student team is also required to deliver 02 Seminars (Power Point Presentations) of 15-20 minutes each, which reflect their learning outcomes. 6. At the end of the term, each student team has to present a synthesis of their work in a final documented report of approximately 10-15 pages.				

	Faculties shall act as facilitators: Observe students as they work on the activity and provide guidance as well as support wherever required.
Term Work (TW):	Term Work evaluation shall be for Total 25 Marks based on the 02 Seminars (50%), Final Report (20%), Weekly Participation and Reporting (30%) and contents covered therein.

Internship Code	Internship Name	Hours/Duration	Credits
INT32	**Internship-II**	**80-120 hrs (2 -3 Weeks)**	

Prerequisite:	Fundamental knowledge of program specific tools, instruments, devices and programming languages etc.
Internship Objectives:	1. To get the exposure to Innovation/IPR/ Entrepreneurship/ Startup initiatives 2. To participate & experience Incubation, Innovation & Business development culture
Internship Outcomes:	**Upon completion of the course, students will be able to:** 1. Learn innovation and entrepreneurial skills to supplement engineering knowledge. 2. Integrate theoretical aspects learned in classes with the practical world 3. Develop an innovative idea to be processed as a start-up

Activity- Innovation/ IPR/ Entrepreneurship	**Supporting Activities to be completed under Internship**
	1. Participation in Innovation related competitions e.g. Hackathons etc.
	2. Awareness & knowledge sessions about Development of new product/Business Plan/Registration of Start-up
	3. Participation in all activities of IIC Cell, E-Cell, NISP, IPR Cell like ● IPR workshop/ ● Leadership Talk ● Idea Design ● Innovation/Business Competition

Term Work Assessment:
Duration to be considered for assessment:
Week Ends/ Semester Break/End of Semester (After ESE & Before Next Term Start)

Guidelines:	1. Batch wise Faculty Supervisor who is the proctor (mentor) of the batch will be allotted as in-charge for the course, at start of the Academic year. 2. Students will submit the participation certificate of the activities to the faculty mentors. 3. For working in cells related activities, Cell coordinator will submit list of actively involved & participated students of each department, semester wise to all department HODs, verified and authenticated by Dean Students Welfare. 4. HODs will circulate the student list to all faculty mentors for consideration of Hours spends under mentioned department activities. 5. Department IIIC Cell coordinator will collect, maintain each student proofs/reports from all faculty mentors, department internship analysis report will be prepared & submitted to Dean, IIIC for AICTE-CII survey data 2. Students will submit evaluation sheet by attaching Xerox copies of all participation/ IPR/ Copyright certificates & faculty mentor will verify it with original copies, for assessment purpose.