

AC – 11.05.2017

Item No. 4.193

UNIVERSITY OF MUMBAI



Revised syllabus (Rev- 2016) from Academic Year 2016 -17

Under

FACULTY OF TECHNOLOGY

Computer Engineering

Second Year with Effect from AY 2017-18

As per **Choice Based Credit and Grading System**

with effect from the AY 2016–17

Co-ordinator, Faculty of Technology's Preamble:

To meet the challenge of ensuring excellence in engineering education, the issue of quality needs to be addressed, debated and taken forward in a systematic manner. Accreditation is the principal means of quality assurance in higher education. The major emphasis of accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Technology of University of Mumbai has taken a lead in incorporating philosophy of outcome based education in the process of curriculum development.

Faculty of Technology, University of Mumbai, in one of its meeting unanimously resolved that, each Board of Studies shall prepare some Program Educational Objectives (PEO's) and give freedom to affiliated Institutes to add few (PEO's). It is also resolved that course objectives and course outcomes are to be clearly defined for each course, so that all faculty members in affiliated institutes understand the depth and approach of course to be taught, which will enhance learner's learning process. It was also resolved that, maximum senior faculty from colleges and experts from industry to be involved while revising the curriculum. I am happy to state that, each Board of studies has adhered to the resolutions passed by Faculty of Technology, and developed curriculum accordingly. In addition to outcome based education, semester based credit and grading system is also introduced to ensure quality of engineering education.

Choice based Credit and Grading system enables a much-required shift in focus from teacher-centric to learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on continuous evaluation which will enhance the quality of education. University of Mumbai has taken a lead in implementing the system through its affiliated Institutes and Faculty of Technology has devised a transparent credit assignment policy and adopted ten points scale to grade learner's performance. Credit assignment for courses is based on 15 weeks teaching learning process, however content of courses is to be taught in 12-13 weeks and remaining 2-3 weeks to be utilized for revision, guest lectures, coverage of content beyond syllabus etc.

Choice based Credit and grading system is implemented from the academic year 2016-17 through optional courses at department and institute level. This will be effective for SE, TE and BE from academic year 2017-18, 2018-19 and 2019-20 respectively.

Dr. S. K. Ukarande
Co-ordinator,
Faculty of Technology,
Member - Academic Council
University of Mumbai, Mumbai

Chairman's Preamble:

Engineering education in India is expanding and is set to increase manifold. The major challenge in the current scenario is to ensure quality to the stakeholders along with expansion. To meet this challenge, the issue of quality needs to be addressed, debated and taken forward in a systematic manner. Accreditation is the principal means of quality assurance in higher education and reflects the fact that in achieving recognition, the institution or program of study is committed and open to external review to meet certain minimum specified standards. The major emphasis of this accreditation process is to measure the outcomes of the program that is being accredited. Program outcomes are essentially a range of skills and knowledge that a student will have at the time of graduation from the program. In line with this Faculty of Technology of University of Mumbai has taken a lead in incorporating the philosophy of outcome based education in the process of curriculum development.

As the Chairman, Board of Studies in Computer Engineering of the University of Mumbai, I am happy to state here that, the Program Educational Objectives for Undergraduate Program were finalized in a brainstorming session, which was attended by more than 85 members from different affiliated Institutes of the University. They are either Heads of Departments or their senior representatives from the Department of Computer Engineering. The Program Educational Objectives finalized for the undergraduate program in Computer Engineering are listed below;

1. To prepare the Learner with a sound foundation in the mathematical, scientific and engineering fundamentals.
2. To motivate the Learner in the art of self-learning and to use modern tools for solving real life problems.
3. To equip the Learner with broad education necessary to understand the impact of Computer Science and Engineering in a global and social context.
4. To encourage, motivate and prepare the Learner's for Lifelong- learning.
5. To inculcate professional and ethical attitude, good leadership qualities and commitment to social responsibilities in the Learner's thought process.

In addition to Program Educational Objectives, for each course of the program, objectives and expected outcomes from a learner's point of view are also included in the curriculum to support the philosophy of outcome based education. I strongly believe that even a small step taken in the right direction will definitely help in providing quality education to the major stakeholders.

Dr. Subhash K. Shinde
Chairman, Board of Studies in Computer Engineering,
University of Mumbai, Mumbai.

Program Structure B.E. Computer Engineering, (Rev. 2016) w.e.f. AY 2017-18 S.

E. Computer Engineering (Semester-III)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract	Tut	Theory	TW/Pract	Tut	Total
CSC301	Applied Mathematics -III	4+1@	-	-	5	-	-	5
CSC302	Digital Logic Design and Analysis	4	-	-	4	-	-	4
CSC303	Discrete Mathematics	3+1@	-	-	4	-	-	4
CSC304	Electronic Circuits and Communication Fundamentals	4	-	-	4	-	-	4
CSC305	Data Structures	4	-	-	4	-	-	4
CSL301	Digital System Lab	-	2	-	-	1	-	1
CSL302	Basic Electronics Lab	-	2	-	-	1	-	1
CSL303	Data structure Lab	-	2	-	-	1	-	1
CSL304	OOPM(Java) Lab	-	2+2*	-	-	2	-	2
Total		21	10	-	21	5	-	26

@ 1 hour to be taken tutorial as class wise.

*2 hours shown as practical's to be taken class wise and other 2 hours to be taken as batch wise

Course Code	Course Name	Examination Scheme								
		Theory					TW	Oral	Oral & Pract	Total
		Internal Assessment			End Sem. Exam	Exam Duration (in Hrs)				
		Test 1	Test 2	Avg.						
CSC301	Applied Mathematics -III	20	20	20	80	3	-	-	-	100
CSC302	Digital Logic Design and Analysis	20	20	20	80	3	-	-	-	100
CSC303	Discrete Structures	20	20	20	80	3	-	-	-	100
CSC304	Electronic Circuits and Communication Fundamentals	20	20	20	80	3	-	-	-	100
CSC305	Data Structures	20	20	20	80	3	--	-	-	100
CSL301	Digital System Lab	-	-	-	-	-	25	--	25	50
CSL302	Basic Electronics Lab	-	-	-	-	-	25	25	---	50
CSL303	Data structure Lab	-	-	-	-	-	25	-	25	50
CSL304	OOPM(Java) Lab	-	-	-	-	-	50	--	50	100
Total		100	100	100	400	-	125	25	100	750

Program Structure B.E. Computer Engineering, (Rev. 2016) w.e.f. AY 2017-18

S. E. Computer Engineering (Semester-IV)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract	Tut	Theory	TW/Pract	Tut	Total
CSC401	Applied Mathematics- IV	4+1@	-	-	5	-	-	5
CSC402	Analysis of Algorithms	4	-	-	4	-	-	4
CSC403	Computer Organization and Architecture	4	-	-	4	-	-	4
CSC404	Computer Graphics	4	-	-	4	-	-	4
CSC405	Operating System	4	-	-	4	-	-	4
CSL401	Analysis of Algorithms Lab	-	2	-	-	1	-	1
CSL402	Computer Graphics Lab	-	2	-	-	1	-	1
CSL403	Processor Architecture Lab	-	2	-	-	1	-	1
CSL404	Operating System Lab	-	2	-	-	1	-	1
CSL405	Open Source Tech Lab	-	2+2*	-	-	2	-	2
Total		21	12	-	21	6	-	27

@ 1 hour to be taken tutorial as class wise .

*2 hours shown as Practical's to be taken class wise and other 2 hours to be taken as batch wise

Course Code	Course Name	Examination Scheme								
		Theory					TW	Oral	Oral & Pract	Total
		Internal Assessment			End Sem. Exam	Exam Duration (in Hrs)				
		Test 1	Test 2	Avg.						
CSC401	Applied Mathematics- IV	20	20	20	80	3	-	-	-	100
CSC402	Analysis of Algorithms	20	20	20	80	3	-	-	-	100
CSC403	Computer Organization and Architecture	20	20	20	80	3	-	-	-	100
CSC404	Computer Graphics	20	20	20	80	3	-	-	-	100
CSC405	Operating System	20	20	20	80	3	--	-	-	100
CSL401	Analysis of Algorithms Lab	-	-	-	-	-	25	--	25	50
CSL402	Computer Graphics Lab	-	-	-	-	-	25	--	25	50
CSL403	Processor Architecture Lab	-	-	-	-	-	25	25	-	50
CSL404	Operating System Lab	-	-	-	-	-	25	-	25	50
CSL405	Open Source Tech Lab	-	-	-	-	-	25	---	25	50
Total		100	100	100	400	-	125	25	100	750

Course Code	Course Name	Credits
CSC301	Applied Mathematics-III	5

Course objectives:

1. To understand the concept of complex variables, C-R equations, harmonic functions and its conjugate and mapping in complex plane.
2. To learn the complex mapping, standard mappings, cross ratios and fixed point.
3. To learn the Laplace Transform, Inverse Laplace Transform of various functions, its application and Z-transform.
4. To understand the concept of Fourier Series, its complex form and enhance the problem solving skill.

Course outcomes: On successful completion of course learner will be able to:

1. Understand complex variable theory, application of harmonic conjugate to get orthogonal trajectories and analytic function.
2. Plot the image of the curve by a complex transformation from z-plane to w-plane.
3. Expand the periodic function by using Fourier series and complex form of Fourier series.
4. Understand the concept of Laplace transform and inverse Laplace transform of various functions and its application to solve ordinary differential equations.
5. Apply the concept of Z- transformation and its inverse of the given sequence.
6. Apply the concept of Correlation and Regression to the engineering problems.

Module No.	Unit No.	Topics	Hrs.
1.0		Laplace Transform	09
	1.1	Laplace Transform of Standard Functions: Introduction, Definition of Laplace transform, Laplace transform of $1, e^{at} \sin(at), \cos(at), \sinh(at), \cosh(at), t^n \operatorname{erf}(t)$, Heavi-side unit step, dirac-delta function, LT of periodic function.	
	1.2	Properties of Laplace Transform: Linearity, first shifting property, second shifting property, multiplication by t^n , division by t , Laplace Transform of derivatives and integrals, change of scale property. (without proof)	
2.0		Inverse Laplace Transform	08
	2.1	Inverse Laplace Transform by Partial fraction method, Convolution theorem	
	2.2	Application to solve initial and boundary value problem involving	

		ordinary differential equations with one dependent variable and constant coefficients.	
3.0		Fourier Series	10
	3.1	Dirichlet's conditions, Fourier series of periodic functions with period 2π and $2L$, Fourier series for even and odd functions.	
	3.2	Half range sine and cosine Fourier series, Parseval's identities (without proof)	
	3.3	Complex form of Fourier series, Orthogonal and Orthonormal set of functions.	
4.0		Complex Variable & mapping	09
	4.1	Functions of a complex variable, Analytic functions, Cauchy-Riemann equations in Cartesian co-ordinates & Polar co-ordinates.	
	4.2	Harmonic functions, Analytic method and Milne Thomson methods to find $f(z)$, Orthogonal trajectories.	
	4.3	Mapping: Conformal mapping, bilinear transformations, cross ratio, fixed points, bilinear transformation of straight lines and circles.	
5.0		Z-transform	06
	5.1	Z-transform of standard functions such as $Z(a^n)$, $Z(n^p)$.	
	5.2	Properties of Z-transform :Linearity, Change of scale, Shifting property, Multiplication of K, Initial and final value, Convolution theorem (without proof)	
	5.3	Inverse Z transform: Binomial Expansion and Method of Partial fraction	
6.0		Correlation & regression, Curve Fitting	10
	6.1	Scattered diagrams, Karl Pearson's coefficient of correlation, covariance, Spearman's Rank correlation(non-repeated and repeated ranks)	
	6.2	Regression coefficient & Lines of Regression.	
	6.3	Fitting of curves: Least square method. Fitting of the straight line $y = a + bx$, parabolic curve $y = a + bx + cx^2$, & exponential curve $y = a b^x$	
		Total	52

Text Books:

1. Higher Engineering Mathematics by Grewal B. S. 38th edition, Khanna Publication 2005.
2. Advanced Engineering Mathematics by Kreyszig E. 9th edition, John Wiley.
3. A Text Book of Applied Mathematics Vol. I & II by P.N.Wartilar &
4. J.N.Wartikar, Pune, Vidyarthi Griha Prakashan., Pune.

Reference Books:

1. Advanced Engg. Mathematics by C. Ray Wylie & Louis Barrett.TMH International Edition.
2. Mathematical Methods of Science and Engineering by Kanti B. Datta, Cengage Learning.
3. Integral Transforms and their Engineering Applications by Dr. B. B. Singh, Synergy Knowledgewar.
4. Laplace Transforms by Murry R. Spieget, Schaun“s out line series-McGraw Hill Publication.

Assessment:**Internal Assessment:**

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

- 1 Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

Course Code	Course Name	Credits
CSC302	Digital Logic Design and Analysis	4

Course Objectives:

1. To introduce the fundamental concepts and methods for design of digital circuits and a pre-requisite for computer organization and architecture, microprocessor systems.
2. To provide the concept of designing Combinational and sequential circuits.
3. To provide basic knowledge of how digital building blocks are described in VHDL.

Course Outcomes: At the end of the course student should be able-

1. To understand different number systems and their conversions.
2. To analyze and minimize Boolean expressions.
3. To design and analyze combinational circuits.
4. To design and analyze sequential circuits
5. To understand the basic concepts of VHDL.
6. To study basics of TTL and CMOS Logic families.

Module	Detailed Content	Hours
1	<p>Number Systems and Codes Introduction to number system and conversions: Binary, Octal, Decimal and Hexadecimal number Systems, Binary arithmetic: addition, subtraction (1's and 2's complement), multiplication and division. Octal and Hexadecimal arithmetic: Addition and Subtraction (7's and 8's complement method for octal) and (15's and 16's complement method for Hexadecimal). Codes: Gray Code, BCD Code, Excess-3 code, ASCII Code. Error Detection and Correction: Hamming codes.</p>	8
2	<p>Boolean Algebra and Logic Gates: Theorems and Properties of Boolean Algebra, Boolean functions, Boolean function reduction using Boolean laws, Canonical forms, Standard SOP and POS form. Basic Digital gates: NOT , AND , OR , NAND , NOR , EXOR , EX-NOR, positive and negative logic, K-map method 2 variable, 3 variable, 4 variable, Don't care condition, Quine-McClusky Method, NAND-NOR Realization.</p>	8
3	<p>Combinational Logic Design: Introduction, Half and Full Adder, Half subtractor Full Subtractor, Four Bit Ripple adder, look ahead carry adder, 4 bit adder subtractor, one digit BCDAdder,Multiplexer,Multiplexertree,Demultiplexer, Demultiplexer tree, Encoders Priority encoder, Decoders, One bit, Two bit , 4-bit Magnitude Comparator, ALU IC 74181.</p>	8

4	<p>Sequential Logic Design: Introduction: SR latch, Concepts of Flip Flops: SR, D, J-K, T, Truth Tables and Excitation Tables of all types, Race around condition, Master Slave J-K Flip Flops, Timing Diagram, Flip-flop conversion, State machines, state diagrams, State table, concept of Moore and Mealy machine.</p> <p>Counters : Design of Asynchronous and Synchronous Counters, Modulus of the Counters, UP- DOWN counter, Shift Registers: SISO, SIPO, PIPO, PISO Bidirectional Shift Register, Universal Shift Register, Ring and twisted ring/Johnson Counter, sequence generator.</p>	15
5	<p>Introduction to VHDL: Introduction: Fundamental building blocks Library, Entity, Architecture, Modeling Styles, Concurrent and sequential statements, simple design examples for combinational circuits and sequential circuits.</p>	6
6	<p>Digital Logic Families: Introduction: Terminologies like Propagation Delay, Power Consumption, Fan in and Fan out , current and voltage parameters, noise margin, with respect to TTL and CMOS Logic and their comparison</p>	3

Text Books:

1. R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill.
2. M. Morris Mano, “Digital Logic and computer Design”, PHI.
3. Norman Balabanian, “Digital Logic Design Principles”, Wiley.
4. J. Bhasker. “VHDL Primer”, Pearson Education.

Reference Books:

1. Donald p Leach, Albert Paul Malvino, “Digital principles and Applications”, Tata McGraw
2. Yarbrough John M. , “Digital Logic Applications and Design “, Cengage Learning.
3. Douglas L. Perry, “VHDL Programming by Example”, Tata McGraw Hill.

Internal Assessment: Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

Course Code	Course Name	Credits
CSC303	Discrete Mathematics	4

Course Objectives:

1. Cultivate clear thinking and creative problem solving.
2. Thoroughly train in the construction and understanding of mathematical proofs. Exercise common mathematical arguments and proof strategies.
3. Thoroughly prepare for the mathematical aspects of other Computer Engineering courses

Course Outcomes: At the end of the course student will be able to

1. Understand the notion of mathematical thinking, mathematical proofs and to apply them in problem solving.
2. Ability to reason logically.
3. Ability to understand relations, Diagraph and lattice..
4. Ability to understand use of functions, graphs and their use in programming applications.
5. Understand use of groups and codes in Encoding-Decoding
6. Apply discrete structures into other computing problems such as formal specification, verification, artificial intelligence, cryptography, Data Analysis and Data Mining etc.

Prerequisite: Basic Mathematics

Sr. No.	Module	Detailed Content	Hours
1	Set Theory	<ul style="list-style-type: none"> • Sets, Venn diagrams, Operations on Sets • Laws of set theory, Power set and Products • Partitions of sets, The Principle of Inclusion and Exclusion 	4
2	Logic	<ul style="list-style-type: none"> • Propositions and logical operations, Truth tables • Equivalence, Implications • Laws of logic, Normal Forms • Predicates and Quantifiers • Mathematical Induction 	8
3	Relations and Functions	<ul style="list-style-type: none"> • Relations, Paths and Digraphs • Properties and types of binary relations • Operations on relations, Closures, Warshall's algorithm • Equivalence and partial ordered relations, • Poset, Hasse diagram and Lattice • Functions: Types of functions - Injective, Surjective and Bijective 	12

		<ul style="list-style-type: none"> • Composition of functions , Identity and Inverse function • Pigeon-hole principle 	
4	Counting	<ul style="list-style-type: none"> • Permutations , Combinations • Elements of Probability, Discrete Probability and Conditional Probability • Generating Functions and Recurrence Relations • Recursive Functions • Introduction to Functional Programming 	6
5	Graphs	<ul style="list-style-type: none"> • Definitions, Paths and circuits: Eulerian and Hamiltonian • Types of graphs, Sub Graphs • Isomorphism of graphs 	6
6	Algebraic Structures and Coding Theory	<ul style="list-style-type: none"> • Algebraic structures with one binary operation: semigroup, monoid and group, Abelian group • Isomorphism, Homomorphism and Automorphism • Cyclic groups, Normal subgroups, • Codes and group codes 	8

Text Books:

1. BernadKolman, 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 BookCompany. Reprinted 2000.
3. K.H.Rosen, Discrete Mathematics and applications, fifth edition 2003, TataMcGraw Hill publishing Company.
4. D.E. Rydeheard University of Manchester , R.M. Burstall, University of Edinburgh “Computational Category Theory”.

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”, TataMcgraw-Hill.
4. Seymour Lipschutz , Marc Lars Lipson,“ Discrete Mathematics” Schaum*sOutline, McGraw Hill Education.

Internal Assessment:

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

**** Tutorial lecture can be conducted for each unit and min 10 problems on the covered unit can be given to the students for practice.**

Course Code	Course Name	Credits
CSC304	Electronic Circuits and Communication Fundamentals	4

Course Objectives:

1. To develop the knowledge of semiconductor devices and circuits, and explain their use in communication applications.
2. To inculcate circuit analysis capabilities in students.
3. To gain knowledge in electronic devices and circuits that is useful in real life applications.
4. To understand the fundamental concepts of electronic communication and their use in computer applications.

Course Outcomes: At the end of the course student should be able

1. To understand the use of semiconductor devices in circuits and analyze them.
2. To understand importance of oscillators and power amplifiers in communication system.
3. To understand basic concepts of operational amplifier and their applications.
4. To understand the fundamental concepts of electronic communication
5. To apply knowledge of electronic devices and circuits to communication applications.
6. To study basic concepts of information theory.

Prerequisite: Basic electrical engineering

Module	Detailed Content	Hours
1	Electronic Circuits: Bipolar junction transistor. Input and Output characteristics, Types of Biasing - Fixed bias, self-bias, voltage divider bias, DC load line and significance, CE amplifier using re model, (Analysis based Numericals)	08
2	Power Amplifiers: Introduction, Class A and Class C power amplifier. Oscillators: Introduction, Barkhausen criteria, Colpitts oscillator and Crystal oscillator	04
3	Electronic Circuits : Operational Amplifier and its applications Op-amp – block diagram, parameters and characteristics, applications- Inverting and Non inverting amplifier, Summing Amplifier(Numerical), Difference amplifier, Basic Integrator and Differentiator, Comparator, Zero Crossing Detector (only theory)	10

4	<p>Communication Fundamentals: Analog Communication</p> <p>Block diagram and elements of analog communication systems, Theory of amplitude modulation and types of AM (Numerical)</p> <p>Generation of DSB SC using diode based balanced modulator, Generation of SSB using phase shift method, Introduction of FM, and its mathematical representation, Statement of Carson's Rule Comparison of AM, FM, Block diagram of AM transmitter (HLM and LLM)</p> <p>Block diagram of AM Superheterodyne receiver.</p>	10
5	<p>Pulse Modulation and Multiplexing.</p> <p>Statement of Sampling Theorem, Generation and detection of PAM, PWM, PPM, PCM, DM and ADM.</p> <p>Principle of TDM using PCM and FDM</p>	10
6	<p>Communication Fundamentals: Information theory.</p> <p>Amount of information, average information, information rate, Statement of Shannon's theorem, channel capacity (Numericals)</p>	06

Text Books:

1. Robert Boylestad, „Electronic Devices and circuit Theory“, Prentice Hall.
2. D Roy Choudhury, „ Linear integrated Circuits“ New Age International Ltd
3. G. Kennedy, B. Davis, S R M Prasanna, „Electronic Communication Systems“, McGraw Hill, 5th Edition.
4. Wayne Tomasi, „Electronic Communication Systems (fundamentals through advanced)“, Pearson Education, 4th Edition.
5. K. Sam Shanmugam,“ Digital and analog communication systems“, Wiley.

Reference Books:

1. Donald Neamen, „Electronic Circuit Analysis and Design“, Tata McGraw Hill, 2nd Edition.
2. K. R. Botkar, „Integrated Circuits“, Khanna Publishers, 9th Edition
3. Simon Haykin, „Digital Communication systems“, Wiley.
4. David Bell, „Electronic Devices and Circuits“, Oxford, 5th Edition.
5. Ramakant A. Gayakwad, „Op-amp and linear integrated circuits“, PHI, 3rd edition.

Internal Assessment

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

Theory Examination:

2. Question paper will comprise of 6 questions, each carrying 20 marks.
5. The students need to solve total 4 questions.
6. Question No.1 will be compulsory and based on entire syllabus.
7. Remaining question (Q.2 to Q.6) will be selected from all the modules.

Course Code	Course Name	Credits
CSC305	Data Structures	4

Course Objectives:

1. To teach various storage mechanisms of data.
2. To design and implement various data structures.
3. To introduce various techniques for representation of the data in the real world.
4. To teach different sorting techniques.
5. To teach different searching techniques.

Course Outcomes:

1. Students will be able to implement various linear and nonlinear data structures.
2. Students will be able to handle operations like insertion, deletion, searching and traversing on various data structures.
3. Students will be able to select appropriate sorting technique for given problem.
4. Students will be able to select appropriate searching technique for given problem.
5. Students will be able to apply the learned concepts in various domains like DBMS and Compiler Construction.
6. Students will be able to choose appropriate data structure for specified problem domain.

Prerequisite: C Programming

Module	Detailed Content	Hours
01	Introduction to Data Structures	03
	Introduction, Types of Data Structures – Linear and Nonlinear, Operations on Data Structures, Concept of ADT, Arrays.	
02	Stack and Queues	10
	Introduction, ADT of Stack, Operations on Stack, Array Implementation of Stack, Applications of Stack – Well form-ness of Parenthesis, Infix to Postfix Conversion and Postfix Evaluation, Recursion. ADT of Queue, Operations on Queue, Array Implementation of Queue, Circular Queue, Priority Queue, Double Ended Queue, Applications of Queue.	
03	Linked List	10
	Introduction, Representation of Linked List, Linked List v/s Array, Implementation of Linked List, Linked Implementation of Stack and Queue, Circular Linked List, Doubly Linked List, Application – Polynomial Representation and Addition.	
04	Trees	12
	Introduction, Tree Terminologies, Binary Tree, Representation, Types of Binary Tree, Binary Tree Traversals, Binary Search Tree, Implementation of Binary Search Tree, Applications – Expression Tree, Huffman Encoding. Search Trees – AVL, B Tree, B+ Tree, Splay Tree and Trie.	

05	Graphs	06
	Introduction, Graph Terminologies, Representation, Graph Traversals – Depth First Search (DFS) and Breadth First Search (BFS), Application – Topological Sorting	
06	Sorting and Searching	07
	Introduction, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort. Linear Search, Binary Search, Hashing – Concept, Hash Functions, Collision Handling Techniques.	

Text Books:

1. Data Structures using C, Reema Thareja, Oxford
2. Data Structures using C and C++, Rajesh K Shukla, Wiley - India
3. Data Structures Using C, Aaron M Tenenbaum, Yedidyah Langsam, Moshe J Augenstein, Pearson
4. Data Structures: A Pseudocode Approach with C, Richard F. Gilberg & Behrouz A., Forouzan, Second Edition, CENGAGE Learning
5. Introduction to Data Structure and Its Applications, JeanPaul Tremblay, P. G. Sorenson

Reference Books:

1. C & Data Structures, Prof. P.S. Deshpande, Prof. O.G. Kakde, DreamTech press.
2. Data Structure Using C, Balagurusamy.
3. Data Structures Using C, ISRD Group, Second Edition, Tata McGraw-Hill.
4. Data Structures, Adapted by: GAV PAI, Schaum's Outlines.

Internal Assessment:

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

- 1 Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules

Lab Code	Lab Name	Credits
CSL301	Digital System Lab	1

Lab Outcome:

1. Understand the basics of various digital components.
2. Understand the principles of design of combinational logic and sequential logic circuits using basic components.
3. Recognize the importance of digital systems in computer architecture.
4. Design and simulate the basic digital circuit.

Description

Experiments with Logic Building Blocks using SSI/MSI, Experiments on Design and/or use Minimization tools. Use of VHDL and simulation in Logic Design. Experiment on design using MSI and/or PLDs tools.

Suggested List of Experiments:

Sr. No.	Title of Experiments
1	To study and verify the truth table of various logic gates using ICs and realize Boolean expressions using gates
2	To realize basic gates using universal gates
4	To realize binary to gray code and gray code to binary converter.
5	To realize parity generator and detector.
6	To realize arithmetic circuits i) Half adder ii) Full adder iii) Half subtractor iv) Full subtractor
7	To realize 2 bit magnitude comparator.
8	To Study multiplexer IC and realization of full adder using multiplexer IC
9	To Study decoder IC and realization of combinational logic using decoder IC
10	Study of flip-flops using IC's
11	To realize asynchronous 3 bit up counter.
12	To realize shift registers using flip flops
13	To realize basic gates using VHDL

14	To realize 4:1 multiplexer using VHDL
15	To realize 4 bit counter using VHDL

Term Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

Oral & Practical exam will be based on the above and CSC302: Digital Logic Design and Analysis syllabus.

Lab Code	Lab Name	Credits
CSL302	Basic Electronics Lab	1

Lab Outcome:

1. Understand the basics of various semiconductor devices, electronic components and instruments.
2. Understand the working of electronic circuits using components
3. Recognize the importance of electronic circuits in electronic communications.
4. Study the fundamental concepts of various modulation methods.

Description

Experiments with semiconductor devices, ICs, electronic component and various measuring instruments. Study experiments on various modulation methods.

Suggested List of Experiments:

Sr. No.	List Experiments
1	Study of electronic components and measuring instruments.
2	Implementation of single stage BJT amplifier.
3	Implementation of oscillators.
4	Implementation of inverting, non inverting amplifier using IC741.
5	Implementation of adder and subtractor using IC 741.
6	Implementation of differentiator using IC741.
7	Implementation of integrator using IC741.
8	Modulation and Demodulation of AM.
9	Study of super heterodyne receiver
10	Modulation and Demodulation PAM.
11	Modulation and Demodulation PWM.
12	Modulation and Demodulation PPM.
13	Modulation and Demodulation PCM.
14	Study of TDM.

15	Study of FDM.
16	SPICE based simulation.

Term Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

Oral exam will be based on the above mentioned experiment list and CSC304: Electronic circuits and communication fundamentals syllabus.

Lab Code	Lab Name	Credits
CSL303	Data Structures Lab	1

Lab outcomes:

1. Students will be able to implement various linear and nonlinear data structures.
2. Students will be able to handle operations like insertion, deletion, searching and traversing on various data structures.

Description: Experiments based on creating and manipulating various data structures.

Suggested Experiments:

Students are required to complete at least 12 experiments.
Star (*) marked experiments are compulsory.

- *1) Array Implementation of Stack.
- *2) Conversion of Infix to Postfix.
- 3) Evaluation of Postfix Expression.
- 4) Check continuity of different types of parenthesis using stack.
- 5) Array Implementation of Queue.

- 7) Array Implementation of Priority Queue
- *8) Implementation of Singly Linked List
- 9) Linked Implementation of Stack
- 10) Linked Implementation of Queue.
- 11) Implementation of Circular Linked List.
- 12) Implementation of Doubly Linked List.
- *13) Implement Binary Search Tree.
- 14) Implementation of Bubble Sort.
- 15) Implementation of Insertion Sort.
- 16) Implementation of Merge Sort.
- *17) Implementation of Quick Sort.
- *18) Implementation of Binary Search.
- 19) Implementation of Hashing.
- 20) Implementation of Depth First Search and Breadth First Search.

Term Work:

1. Term work should consist of at least 10 experiments.
2. Journal must include at least 2 assignments.
3. A case study should be conducted using a Mini Project by taking a good problem definition and complete the following phases.
 - a. Decomposing the problem into modules
 - b. Identifying the best suited data structure for solving the sub problems with justification
 - c. Define algorithms for various identified functions
 - d. Implement the modules
4. The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
5. Term Work:
Total 25 Marks = (Experiments: 10 mark + Mini Project: 05 mark + Assignments: 05 mark)

Practical and oral examination will be based on the above syllabus.

Lab Code	Lab Name	Credits
CSL304	OOPM (JAVA) Lab	2

Course Objective:

1. To learn the object oriented programming concepts.
2. To study various java programming concept like multithreading, exception handling, packages etc.
3. To explain components of GUI based programming.

Course Outcomes: At the end of the course Student should be able:

1. To apply fundamental programming constructs.
2. To illustrate the concept of packages, classes and objects.
3. To elaborate the concept of strings, arrays and vectors.
4. To implement the concept of inheritance and interfaces.
5. To implement the notion of exception handling and multithreading.
6. To develop GUI based application.

Prerequisite: Structured Programming Approach

Sr. No.	Module	Detailed Content	Hours
1	Introduction to Object Oriented Programming	1.1 OOP Concepts: Object, Class, Encapsulation, Abstraction, Inheritance, Polymorphism. 1.2 Features of Java, JVM 1.3 Basic Constructs/Notions: Constants, variables and data types, Operators and Expressions, Revision of Branching and looping	02
2	Classes, Object and Packages	2.1 Class, Object, Method. 2.2 Constructor, Static members and methods 2.3 Passing and returning Objects 2.4 Method Overloading 2.5 Packages in java, creating user defined packages, access specifiers.	05
3	Array, String and Vector	3.1 Arrays, Strings, String Buffer 3.2 Wrapper classes, Vector	04
4	Inheritance and Interface	4.1 Types of Inheritance, super keyword, Method Overriding, abstract class and abstract method, final keyword, 4.2 Implementing interfaces, extending interfaces	03
5	Exception Handling and Multithreading	5.1 Error vs Exception, try, catch, finally, throw, throws, creating own exception 5.2 Thread lifecycle, Thread class methods, creating threads, Synchronization	04
6	GUI programming in JAVA	6.1 Applet: Applet life cycle, Creating applets, Graphics class methods, Font and Color class, parameter passing.	08

		6.2 Event Handling: Event classes and event listener 6.3 Introduction to AWT: Working with windows, Using AWT controls- push Buttons, Label, Text Fields, Text Area, Check Box, and Radio Buttons. 6.4 Programming using JDBC: Introduction to JDBC, JDBC Drivers & Architecture.	
--	--	---	--

Text books:

1. Herbert Schildt, „JAVA: The Complete Reference“, Ninth Edition, Oracle Press.
2. Sachin Malhotra and Saurabh Chaudhary, “Programming in Java”, Oxford University Press, 2010

Reference Books:

1. Ivor Horton, „Beginning JAVA“, Wiley India.
2. DietalandDietal, „Java: How to Program“, 8/e, PHI
3. „JAVA Programming“, Black Book, Dreamtech Press.
4. „Learn to Master Java programming“, Staredusolutions

Digital Material:

1. www.nptelvideos.in
2. www.w3schools.com
3. <http://spoken-tutorial.org>
4. www.staredusolutions.org

Suggested List of Programming Assignments/Laboratory Work:

1.	Program on various ways to accept data through keyboard and unsigned right shift operator.
2.	Program on branching, looping, labelled break and labelled continue.
3.	Program to create class with members and methods, accept and display details for single object.
4.	Program on constructor and constructor overloading
5.	Program on method overloading
6.	Program on passing object as argument and returning object
7.	Program on creating user defined package
8.	Program on 1D array
9.	Program on 2D array

10.	Program on String
11.	Program on StringBuffer
12.	Program on Vector
13.	Program on single and multilevel inheritance (Use super keyword)
14.	Program on abstract class
15.	Program on interface demonstrating concept of multiple inheritance
16.	Program on dynamic method dispatch using base class and interface reference.
17.	Program to demonstrate try, catch, throw, throws and finally.
18.	Program to demonstrate user defined exception
19.	Program on multithreading
20.	Program on concept of synchronization
21.	Program on Applet to demonstrate Graphics, Font and Color class.
22.	Program on passing parameters to applets
23.	Program to create GUI application without event handling using AWT controls
24.	Program to create GUI application with event handling using AWT controls
25.	Mini Project based on content of the syllabus. (Group of 2-3 students)

Term Work:

Students will submit term work in the form of journal that will include:

1. At least 16-18 programs and mini project
2. Two assignments covering whole syllabus

Term Work: 50 Marks (Total Marks) = 20 marks (Experiments) +
20 marks (Mini Project) +
05 marks (Assignments) +
05 marks (Attendance)

Practical and oral examination will be based on suggested practical list and entire syllabus.

Course Code	Course Name	Credits
CSC401	Applied Mathematics-IV	5

Course Objectives: The objectives of this course are to teach the students:

1. Matrix theory, and its application to find the matrix function. Present methods of computing and using Eigen values and Eigen vectors.
2. Set up and directly evaluate contour integrals Cauchy's integral theorem and formula in basic and extended form. Present Taylor and Laurent's series to find singularities zero's and poles also presents residues theory
3. Theory of probability, Baye's Theorem, Expectation and Moments and its application.
4. Probability distribution such as Binomial, Poisson and Normal distribution with their properties.
5. Sampling theory and its application for small and large sample and Optimization techniques.

Course Outcomes:

1. Students in this course will be able to apply the method of solving complex integration, computing residues & evaluate various contour integrals.
2. Demonstrate ability to manipulate matrices and compute Eigen values and Eigen vectors.
3. Apply the concept of probability distribution to the engineering problems.
4. Apply the concept of sampling theory to the engineering problems.
5. Use matrix algebra with its specific rules to solve the system of linear equation, using concept of Eigen value and Eigen vector to the engineering problems.
6. Apply the concept of Linear & Non-Linear Programming Problem to the engineering problems.

Module No.	Unit No.	Topics	Hrs.
1.0		Complex Integration	10
	1.1	Complex Integration – Line Integral, Cauchy's Integral theorem for simply connected regions, Cauchy's Integral formula(without proof)	
	1.2	Taylor's and Laurent's series (without proof)	
	1.3	Zeros, poles of $f(z)$, Residues, Cauchy's Residue theorem.	
	1.4	Applications of Residue theorem to evaluate Integrals of the type $2 \int_0^\pi f(\cos \theta, \sin \theta) d\theta, \int_{-\infty}^{\infty} f(x) dx$	
2.0		Matrices	10
	2.1	Eigen values and Eigen vectors.	
	2.2	Cayley-Hamilton theorem(without proof)	
	2.3	Similar matrices, diagonalisable matrix.	
	2.4	Derogatory and non-derogatory matrices, Functions of square matrix.	

3.0		Probability	10
	3.1	Baye's Theorem	
	3.2	Random Variables: Discrete & continuous random variables, expectation, Variance, Probability Density Function & Cumulative Density Function.	
	3.3	Moments & Moment generating function.	
	3.4	Probability distribution: Binomial distribution, Poisson & Normal distribution. (For detail study)	
4.0		Sampling Theory (Large Sample test)	06
	4.1	Sampling Distribution, Test of Hypothesis, Level of significance, Critical region, One Tailed and Two Tailed test,	
	4.2	Test of significant for Large Samples:-Means of the samples and test of significant of means of two large samples.	
5.0		Sampling Theory (Small Sample test)	06
	5.1	Test of significant for small samples:- Students t- distribution for dependent and independent samples	
	5.2	Chi square test:- Test of goodness of fit and independence of attributes,Contingency table.	
6.0		Mathematical Programming	10
	6.1	Types of solution, Standard and Canonical form of LPP, Basic and feasible solutions, simplex method.	
	6.2	Artificial variables, Big –M method (method of penalty).	
	6.3	Duality and Dual simplex method.	
	6.4	Non Linear Programming Problems with equality constrains and inequality Constrains (two or three variables with one constrains) (No formulation, No Graphical method).	
		Total	52

Text Books:

1. Higher Engineering Mathematics by Grewal B. S. 38th edition, Khanna Publication 2005.
2. Operation Research by Hira & Gupta,S Chand.
3. A Text Book of Applied Mathematics Vol. I & II by P.N.Wartilar &
4. J.N.Wartikar, Pune, Vidyarthi Griha Prakashan., Pune.
5. Probability and Statistics for Engineering, Dr. J Ravichandran, Wiley-India.

Reference Books:

1. Probability & Statistics with reliability by Kishor s. Trivedi, Wiley India.
2. Advanced Engg. Mathematics by C. Ray Wylie & Louis Barrett.TMH International Edition.
3. Mathematical Methods of Science and Engineering by Kanti B. Datta, Cengage Learning.
4. Advanced Engineering Mathematics by Kreyszig E. 9th edition, John Wiley.
5. Operations Research by S.D. Sharma Kedar Nath, Ram Nath & Co. Meerat.
6. Engineering optimization (Theory and Practice) by Singiresu S.Rao, New Age International publication.

Assessment:**Internal Assessment:**

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

In question paper, weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Course Code	Course Name	Credits
CSC402	Analysis of Algorithms	4

Course Objectives:

4. To provide mathematical approach for Analysis of Algorithms
5. To solve problems using various strategies
6. To analyse strategies for solving problems not solvable in polynomial time.

Course Outcomes: At the end of the course student will be able to

1. Analyze the running time and space complexity of algorithms.
2. Describe, apply and analyze the complexity of divide and conquer strategy.
3. Describe, apply and analyze the complexity of greedy strategy.
4. Describe, apply and analyze the complexity of dynamic programming strategy.
5. Explain and apply backtracking, branch and bound and string matching techniques to deal with some hard problems.
6. Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete.

Prerequisites: Students should be familiar with concepts of Data structure and discrete structures.

Module	Detailed Content	Hours
1	<p>Introduction to analysis of algorithm Performance analysis , space and time complexity Growth of function – Big –Oh ,Omega , Theta notation Mathematical background for algorithm analysis, Analysis of selection sort , insertion sort.</p> <p>Recurrences: -The substitution method -Recursion tree method -Master method</p> <p>Divide and Conquer Approach: General method Analysis of Merge sort, Analysis of Quick sort, Analysis of Binary search, Finding minimum and maximum algorithm and analysis, Strassen’s matrix multiplication</p>	12
2	<p>Dynamic Programming Approach: General Method Multistage graphs single source shortest path all pair shortest path Assembly-line scheduling 0/1 knapsack Travelling salesman problem Longest common subsequence</p>	08
3	<p>Greedy Method Approach:</p>	06

	General Method Single source shortest path Knapsack problem Job sequencing with deadlines Minimum cost spanning trees-Kruskal and prim's algorithm Optimal storage on tapes	
4	Backtracking and Branch-and-bound: General Method 8 queen problem(N-queen problem) Sum of subsets Graph coloring 15 puzzle problem, Travelling salesman problem.	08
5	String Matching Algorithms: The naïve string matching Algorithms The Rabin Karp algorithm String matching with finite automata The knuth-Morris-Pratt algorithm	06
6	Non-deterministic polynomial algorithms: Polynomial time, Polynomial time verification NP Completeness and reducibility NP Completeness proofs Vertex Cover Problems Clique Problems	08

Text Books:

1. T.H.coreman , C.E. Leiserson,R.L. Rivest, and C. Stein, "Introduction to algorithms", 2nd edition , PHI publication 2005.
2. Ellis horowitz , Sartaj Sahni , S. Rajsekar. "Fundamentals of computer algorithms" University Press

Reference Books:

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw- Hill Edition.
2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI.
3. John Kleinberg, Eva Tardos, "Algorithm Design", Pearson.
4. Michael T. Goodrich, Roberto Tamassia, "Algorithm Design", Wiley Publication.

Assessment:

Internal Assessment:

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules..

Course Code	Course Name	Credit
CSC403	Computer Organization and Architecture	4

Course Objectives:

1. To have a thorough understanding of the basic structure and operation of a digital computer.
2. To discuss in detail the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
3. To study the different ways of communicating with I/O devices and standard I/O interfaces.
4. To study the hierarchical memory system including cache memories and virtual memory.

Course Outcomes: At the end of the course student should be able-

1. To describe basic structure of the computer system.
2. To demonstrate the arithmetic algorithms for solving ALU operations.
3. To describe instruction level parallelism and hazards in typical processor pipelines.
4. To describe superscalar architectures, multi-core architecture and their advantages
5. To demonstrate the memory mapping techniques.
6. To Identify various types of buses, interrupts and I/O operations in a computer system

Prerequisite: Digital Logic Design and Application

Sr. No.	Module	Detailed Content	Hours
1	<u>Introduction</u>	<u>Overview of Computer Architecture & Organization</u> <ul style="list-style-type: none"> • Introduction • Basic organization of computer • Block level description of the functional units. <u>Data Representation and Arithmetic Algorithms:</u> <ul style="list-style-type: none"> • Integer Data computation: Addition, Subtraction. Multiplication: unsigned multiplication, Booth's algorithm. • Division of integers: Restoring and non restoring division • Floating point representation. IEEE 754 floating point number representation. • Floating point arithmetic: Addition, Subtraction, Multiplication, Division 	08
2	<u>Processor Organization and Architecture</u>	<ul style="list-style-type: none"> • Von Neumann model, Harvard Architecture • Register Organization, Instruction formats, addressing modes, instruction cycle. Instruction interpretation and sequencing. • ALU and Shifters • Basic pipelined datapath and control, Data dependences, data hazards, Branch hazards, delayed branches, branch prediction • Performance measures – CPI, speedup, efficiency, throughput and Amdahl's law 	10

3	<u>Control Unit Design</u>	<ul style="list-style-type: none"> • Hardwired control unit design methods: State table, delay element, sequence counter with examples like control unit for multiplication and division • Microprogrammed control Unit: Microinstruction sequencing and execution. Micro operations, Wilkie's microprogrammed Control Unit, Examples on microprograms 	08
4	<u>Memory Organization</u>	<ul style="list-style-type: none"> • Classifications of primary and secondary memories. Types of RAM (SRAM, DRAM, SDRAM, DDR, SSD) and ROM, Characteristics of memory, Memory hierarchy: cost and performance measurement. • Virtual Memory: Concept, Segmentation and Paging, Address translation mechanism. • Interleaved and Associative memory. • Cache memory Concepts, Locality of reference, design problems based on mapping techniques. Cache Coherency, Write Policies 	12
5	<u>I/O Organization and Peripherals</u>	<ul style="list-style-type: none"> • Common I/O device types and characteristics • Types of data transfer techniques: Programmed I/O, Interrupt driven I/O and DMA. • Introduction to buses, Bus arbitration and multiple bus hierarchy • Interrupt types, Interrupts handling 	06
6	<u>Advanced Processor Principles</u>	<ul style="list-style-type: none"> • Introduction to parallel processing, Flynn's Classification • Concepts of superscalar architecture, out-of-order execution, speculative execution, multithreaded processor, VLIW, data flow computing. • Introduction to Multi-core processor architecture 	08

Text Books:

1. William Stallings, "Computer Organization and Architecture: Designing for Performance", Pearson Publication, 10th Edition, 2013
2. John P. Hayes, "Computer Architecture and Organization", McGraw-Hill, 1988
3. B. Govindarajulu, "Computer Architecture and Organization: Design Principles and Applications", Second Edition, McGraw-Hill (India),

Reference Books:

1. Andrew S. Tanenbaum "Structured Computer Organization", Pearson, Sixth Edition
2. Morris Mano. "Computer System Architecture" Pearson Publication, 3rd Edition, 2007
3. Kai Hwang, Fayé Alayé Briggs. "Computer architecture and parallel processing", McGraw-Hill
4. P. Pal Chaudhuri. "Computer Organization and Design" Prentice Hall India, 2004
5. Dr. M. Usha, T.S. Shrikant. "Computer System Architecture and Organization" Wiley India, 2014.

Internal Assessment:

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

Course Code	Course Name	Credits
CSC404	Computer Graphics	4

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 like Image Processing or Computer Vision or Virtual Reality and professional avenues in the field of Computer Graphics.

Course Outcomes : At the end of the course , the students should be able to

- 1 Understand the basic concepts of Computer Graphics.
- 2 Demonstrate various algorithms for scan conversion and filling of basic objects and their comparative analysis.
- 3 Apply geometric transformations, viewing and clipping on graphical objects.
- 4 Explore solid model representation techniques and projections.
- 5 Understand visible surface detection techniques and illumination models.

Prerequisite: Knowledge of C Programming, Basic Data Structures and Mathematics.

Module No	Detail Syllabus	Hours
1	<p>Introduction and Overview of Graphics System:</p> <ul style="list-style-type: none"> • Definition and Representative uses of computer graphics, classification of application areas, Overview of coordinate systems ,definition of scan conversion, rasterization and rendering. • Raster scan & random scan displays, Flat Panel displays like LCD and LED , architecture of raster graphics system with display processor, architecture of random scan systems. 	03
2	<p>Output Primitives :</p> <ul style="list-style-type: none"> • 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) • Aliasing , Antialiasing techniques like Pre and post filtering , super sampling , and pixel phasing). • Filled Area Primitive: Scan line Polygon Fill algorithm, Inside outside tests, Boundary Fill and Flood fill algorithm. 	12
3	<p>Two Dimensional Geometric Transformations</p> <ul style="list-style-type: none"> • Basic transformations : Translation , Scaling , Rotation • Matrix representation and Homogeneous Coordinates • Composite transformation • Other transformations : Reflection and Shear • Raster method for transformation. 	06

4	Two Dimensional Viewing and Clipping <ul style="list-style-type: none"> • Viewing transformation pipeline and Window to Viewport coordinate transformation • Clipping operations – Point clipping , Line clipping algorithms : Cohen – Sutherland , Midpoint subdivision , Liang – Barsky , Polygon Clipping Algorithms : Sutherland – Hodgeman, Weiler – Atherton. 	08
5	Three Dimensional Object Representations , Geometric Transformations and 3D Viewing <ul style="list-style-type: none"> • Boundary Representation and Space partitioning representation: Polygon Surfaces , Bezier Curve , Bezier Surface , B-Spline Curve , Sweep Representation, Constructive Solid Geometry ,Octree, Fractal-Geometry : Fractal Dimension, Koch Curve. • 3D Transformations :Translation, Rotation , Scaling and Reflection. • Composite transformations :Rotation about an arbitrary axis • 3D transformation pipeline • Projections – Parallel , Perspective.(Matrix Representation) • 3D clipping. 	12
6	Visible Surface Detection <ul style="list-style-type: none"> • Classification of Visible Surface Detection algorithm • Back Surface detection method • Depth Buffer method • Depth Sorting method • Scan line method • Area Subdivision method 	04
7	Illumination Models and Surface Rendering <ul style="list-style-type: none"> • Basic Illumination Models : Diffused reflection, Phong Specular reflection Model • Halftone and Dithering techniques • Polygon Rendering :Constant shading , Gouraud Shading , Phong Shading. 	03

Text Books:

1. “Computer Graphics” C version by Hearn & Baker, 2nd Edition, Pearson
2. “Computer Graphics Principles and Practice in C , 2nd Edition ,James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, Pearson .
3. “Computer Graphics”, by Rajesh K. Maurya, Wiley India Publication.
4. “Computer Graphics “ , by Samit Bhattacharya , Oxford Publication.

Reference Books:

1. “Procedural Elements for Computer Graphics “ by D. Rogers , Tata McGraw-Hill Publications.
2. “Computer Graphics” , by Zhigang Xiang , Roy Plastock , Schaum“s Outlines McGraw-Hill Education
3. “Computer Graphics using OpenGL” , by F.S.Hill , Jr. ,Third edition, Pearson Publications.

Assessment:**Internal Assessment:**

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

Course Code	Course Name	Credits
CSC405	Operating System	4

Course Objectives:

1. To introduce basic concepts and functions of operating systems.
2. To understand the concept of process, thread and resource management.
3. To understand the concepts of process synchronization and deadlock.
4. To understand various Memory, I/O and File management techniques.

Course Outcomes: At the end of the course student should be able to

1. Understand role of Operating System in terms of process, memory, file and I/O management.
2. Apply and analyse the concept of a process, thread, mutual exclusion and deadlock.
3. Evaluate performance of process scheduling algorithms and IPC.
4. Apply and analyse the concepts of memory management techniques.
5. Evaluate the performance of memory allocation and replacement techniques.
6. Apply and analyze different techniques of file and I/O management.

Prerequisite: Computer Organization & Architecture

Sr No	Module	Detailed Content	Hours
1	Operating System Overview	Operating System Objectives and Functions, The Evolution of Operating Systems, OS Design Considerations for Multiprocessor and Multicore architectures, Operating system structures, System Calls, Linux Kernel and Shell.	8 hrs
2	Process Concept and Scheduling	<p>Process: Concept of a Process, Process States, Process Description, Process Control Block, Operations on Processes.</p> <p>Threads: Definition and Types, Concept of Multithreading, Multicore processors and threads.</p> <p>Scheduling: Uniprocessor Scheduling - Types of Scheduling: Preemptive and, Non-preemptive, Scheduling Algorithms: FCFS, SJF, SRTN, Priority based, Round Robin, Multilevel Queue scheduling. Introduction to Thread Scheduling, Multiprocessor Scheduling and Linux Scheduling.</p>	8 hrs

3	Synchronization and Deadlocks	<p>Concurrency: Principles of Concurrency, Inter-Process Communication, Process/Thread Synchronization.</p> <p>Mutual Exclusion: Requirements, Hardware Support, Operating System Support (Semaphores and Mutex), Programming Language Support (Monitors), Classical synchronization problems: Readers/Writers Problem, Producer and Consumer problem.</p> <p>Principles of Deadlock: Conditions and Resource Allocation Graphs, Deadlock Prevention, Deadlock Avoidance: Banker's Algorithm for Single & Multiple Resources, Deadlock Detection and Recovery. Dining Philosophers Problem.</p>	12 hrs
4	Memory Management	<p>Memory Management: Memory Management Requirements, Memory Partitioning: Fixed Partitioning, Dynamic Partitioning, Memory Allocation Strategies: Best-Fit, First Fit, Worst Fit, Next Fit, Buddy System, Relocation. Paging, Segmentation.</p> <p>Virtual Memory: Hardware and Control Structures, Demand Paging, Structure of Page Tables, Copy on Write, Page Replacement Strategies: FIFO, Optimal, LRU, LFU, Approximation, Counting Based. Allocation of frames, Thrashing.</p>	8 hrs
5	File Management	<p>File Management: Overview, File Organization and Access, File Directories, File Sharing, Secondary Storage Management, Linux Virtual File System.</p>	6 hrs
6	Input /Output Management	<p>I/O Management and Disk Scheduling: I/O Devices, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling algorithm: FCFS, SSTF, SCAN, CSCAN, LOOK, C-LOOK. Disk Management, Disk Cache, Linux I/O.</p>	6 hrs

Text Books:

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 8th Edition, 2014, ISBN-10: 0133805913 • ISBN-13: 9780133805918 .
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, John Wiley & Sons , Inc., 9th Edition, 2016, ISBN 978-81-265-5427-0
3. Andrew Tannenbaum, Operating System Design and Implementation, Pearson, 3rd Edition.
4. D.M Dhamdhere, Operating Systems: A Concept Based Approach, Mc-Graw Hill

Reference Books:

1. Maurice J. Bach, “Design of UNIX Operating System”, PHI
2. Achyut Godbole and Atul Kahate, Operating Systems, Mc Graw Hill Education, 3rd Edition
3. The Linux Kernel Book, Remy Card, Eric Dumas, Frank Mevel, Wiley Publications.

Assessment:**Internal Assessment:**

Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
4. Remaining question (Q.2 to Q.6) will be selected from all the modules.

Lab Code	Lab Name	Credit
CSL401	Analysis of Algorithms Lab	1

Lab outcomes: At the end of the course student will be able to

1. Analyze the complexities of various problems in different domains.
2. Prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains.
3. Develop the efficient algorithms for the new problem with suitable designing techniques.
4. Implement the algorithms using different strategies.

Prerequisites: Students should be familiar with concepts of Data structure and Discrete structures.

Description:

Minimum 2 experiments should be implemented using any language on each algorithm design strategy (Divide and conquer, dynamic programming, Greedy method, backtracking and branch & bound, string matching).

Suggested Laboratory Experiments:

Sr. No.	Module Name	Suggested Experiment List
1	Introduction to analysis of algorithm Divide and Conquer Approach	Selection sort , insertion sort. Merge sort, Quick sort, Binary search.
2	Dynamic Programming Approach	Multistage graphs, single source shortest path, all pair shortest path, 0/1 knapsack, Travelling salesman problem, Longest common subsequence.
3	Greedy Method Approach	Single source shortest path, Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees-Kruskal and prim's algorithm, Optimal storage on tapes.
4	Backtracking and Branch-and-bound	8 queen problem (N-queen problem), Sum of subsets, Graph coloring, 15 puzzle problem, Travelling salesman problem.

5	String Matching Algorithms	The naïve string matching Algorithms, The Rabin Karp algorithm, String matching with finite automata, The knuth-Morris-Pratt algorithm.
6	Any two Experiments	This will involve implementation of two algorithms for problems beyond the scope of syllabus. The exact set of algorithms to implement is to be decided by the course instructor.

Text Books:

1. T.H.Coreman , C.E. Leiserson,R.L. Rivest, and C. Stein, “Introduction to algorithms”, 2nd edition , PHI publication 2005.
2. Ellis horowitz , sartaj Sahni , s. Rajsekar. “Fundamentals of computer algorithms” University Press

Reference Books:

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, “Algorithms”, Tata McGraw- Hill Edition.
2. S. K. Basu, “Design Methods and Analysis of Algorithm”, PHI.
3. Dana Vrajittoru and William Knight, “Practical Analysis of Algorithms”, Springer 2014th Edition.

Term Work:

Laboratory work must contain implementation of minimum 10 experiments. The final certification and acceptance of term work ensures the satisfactory performance of laboratory work and minimum passing marks in term work. The 25 marks of the term work should be divided as below:
25 Marks (total marks) = 15 Marks Lab. Experiments + 05 Marks Assignments (based on theory syllabus) + 05 (Attendance: theory + practical)

Oral & Practical Exam will be based on the experiments implemented in the Laboratory.

Lab Code	Lab Title	Credit
CSL402	Computer Graphics Lab	1

Lab Objectives

- 1 To emphasize on implementation aspect of Computer Graphics Algorithm.
- 2 To prepare students for advanced areas like Animation, image processing ,virtual reality etc

Lab Outcomes : At the end of the course , the students should be able to

- 1 Explore the working principle, utility of various input/ output devices and graphical tools.
- 2 Implement various output and filled area primitive algorithms using C/ OpenGL
- 3 Apply transformation and clipping algorithms on graphical objects.
- 4 Implementation of curve and fractal generation.
- 5 Develop a Graphical application based on learned concept.

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++/ Open GL

List of Desirable Experiments:

1. Study and apply basic opengl functions to draw basic primitives. (*)
2. Implement sierpinsky gasket using openGL.
3. Implement DDA Line Drawing algorithms and Bresenham algorithm(*)
4. Implement midpoint Circle algorithm(*)
5. Implement midpoint Ellipse algorithm
6. Implement Area Filling Algorithm: Boundary Fill, Flood Fill ,Scan line Polygon Fill (*)
7. Implement Curve : Bezier for n control points , B Spline (Uniform) (atleast one)(*)
8. Implement Fractal (Koch Curve)
9. Character Generation : Bit Map method and Stroke Method
10. Implement 2D Transformations: Translation, Scaling, Rotation, Reflection, Shear.(*)
11. Implement Line Clipping Algorithm: Cohen Sutherland / Liang Barsky.(*)
12. Implement polygon clipping algorithm(atleast one)
13. Program to represent a 3D object using polygon surfaces and then perform 3D transformation.
14. Program to perform projection of a 3D object on Projection Plane : Parallel and Perspective.(*)

Term Work

1. Term work should consist of at least 12 experiments. (*) → Practical to be covered necessarily
2. Journal must include at least 2 assignments.
3. Mini Project to perform using C / OpenGL. Possible Ideas:
 - a. Animation using multiple object
 - b. Graphics editor with following features :
 - *Draw basic geometrical entities; apply geometrical transformations, Area filling, Clipping against Clip window, displaying the text, displaying bar / line graphs , pie charts etc.

The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.

Term Work: 25 Marks (Total) = 10 Marks (Experiments)
+ 5 Marks (Mini Project)
+ 5 Marks (Assignments)
+ 5 Marks (Theory + Practical Attendance).

Oral & Practical exam will be based on the above content and CSC404: Computer Graphics.

Lab Code	Lab Name	Credit
CSL403	Processor Architecture Lab	1

Lab Objectives:

1. To implement the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
2. To study the different ways of communicating with I/O devices and standard I/O interfaces.
3. To design memory subsystem including cache memory
4. To have through understanding of various computer buses

Lab Outcomes: At the end of the course student should be

1. Assemble personal computer
2. Design the basic building blocks of a computer: arithmetic-logic unit, registers, central processing unit, and memory.
3. Implement various algorithms like Booth's algorithm for arithmetic operations
4. Describe various I/O buses with merits and demerits.

Prerequisite: Digital Logic Design and Applications

Content:

Sr. No.	Module	Detailed Content
1	Overview of Computer Architecture & Organization	<ul style="list-style-type: none"> • Computer Anatomy- Memory, Ports, Motherboard and add-on cards • Dismantling and assembling PC
2	Programs on Data Representation and Arithmetic	<ul style="list-style-type: none"> • Ripple carry adder, Carry look-ahead adder, registers, Multiplication • Booths Algorithm, Restoring and Non restoring Division
3	Processor Organization and Architecture	<ul style="list-style-type: none"> • ALU Design, CPU Design • Case Study on multi-core Processors
4	Memory Organization	<ul style="list-style-type: none"> • Memory design, Cache Memory design
5	I/O Organization and Interrupts	<ul style="list-style-type: none"> • Case study on buses like ISA, PCI, USB etc • Interrupt handling using C/Java Programming

Digital Material:

- **Manual to use the simulator for computer organization and architecture.** Developed by the Department of CSE, IIT kharagpur (<http://cse10-iitkgp.virtual-labs.ac.in/>)

Books:

1. William Stallings, “Computer Organization and Architecture: Designing for Performance”, Pearson Publication, 10th Edition, 2013
2. B. Govindarajulu, “Computer Architecture and Organization: Design Principles and Applications”, Second Edition, McGraw-Hill (India),
6. Andrew S. Tanenbaum “Structured Computer Organization”, Pearson, Sixth Edition
7. Morris Mano. “Computer System Architecture” Pearson Publication, 3rd Edition, 2007
8. Kai Hwang, Fayé Alayé Briggs. “Computer architecture and parallel processing”, McGraw-Hill
9. P. Pal Chaudhuri. “Computer Organization and Design” Prentice Hall India, 2004
10. Dr. M. Usha, T.S. Shrikant. “Computer System Architecture and Organization” Wiley India, 2014.

Term Work

Term work should consist of at least 10-12 experiments and 3-4 assignments based on above content and CSC403: Computer Organization and Architecture

The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.

Term Work: 25 Marks (Total) = 10 Marks (Experiments)
+ 5 Marks (Mini Project)
+ 5 Marks (Assignments)
+ 5 Marks (Theory + Practical Attendance).

Oral exam will be based on the above content and CSC403: Computer Organization and Architecture.

Lab Code	Lab Name	Credit
CSL404	Operating System Lab	1

Lab Outcome:

1. Understand basic operating system commands.
2. Understand and explore various system calls.
3. Write shell scripts and shell commands using kernel APIs.
4. Implement and analyze different process scheduling algorithms
5. Implement and analyze different memory management algorithms.
6. Evaluate process management techniques and deadlock handling using simulator.

Descriptions:

Sr. No	Contents
1	Explore the internal commands of linux like ls, chdir, mkdir, chown, chmod, chgrp, ps etc
2	Write shell scripts to do the following: <ul style="list-style-type: none"> ➤ Display top 10 processes in descending order ➤ Display processes with highest memory usage. ➤ Display current logged in user and logname. ➤ Display current shell, home directory, operating system type, current path setting, current working directory. ➤ Display OS version, release number, kernel version. ➤ Illustrate the use of sort, grep, awk, etc.
3	a) Create a child process in Linux using the fork system call. From the child process obtain the process ID of both child and parent by using getpid and getppid system call. Explore wait and waitpid before termination of process. b) Explore the following system calls: open, read, write, close, getpid, setpid, getuid, getgid, getegid, geteuid.
4	Implement basic commands of linux like ls, cp, mv and others using kernel APIs.
5	Write a program to implement any two CPU scheduling algorithms like FCFS, SJF, Round Robin etc.
6	Write a program to implement dynamic partitioning placement algorithms i.e Best Fit, First-Fit, Worst-Fit etc
7	Write a program to implement various page replacement policies.
8	Using the CPU-OS simulator analyze and synthesize the following: <ol style="list-style-type: none"> a. Process Scheduling algorithms. b. Thread creation and synchronization. c. Deadlock prevention and avoidance.

Digital Materials:

1. Download the CPU-OS simulator along with related tutorials from <http://www.teach-sim.com>
2. <http://www.teach-sim.com>

Books:

1. Linux Kernel Book, by Remy Card, Eric Dumas, Frank Mevel, Wiley India
2. Unix Concepts and Applications, Sumitabha Das, McGraw Hill.

Term Work:

- Term work should consist of at least 12 experiments and 2-3 assignments on above content.
- The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
- Term Work: 25 Marks (Total) = 10 Marks (Experiments)
+ 5 Marks (Mini Project)
+ 5 Marks (Assignments)
+ 5 Marks (Theory + Practical Attendance).

Oral & Practical exam will be based on the above content and CSC405: Operating system syllabus.

Lab Code	Lab Name	Credit
CSL405	Open Source Technology Lab	2

Course Outcomes:

1. To understand basic concepts in python and perl.
2. To explore contents of files, directories and text processing with python
3. To develop program for data structure using built in functions in python.
4. To explore django web framework for developing python based web application.
5. To understand file handling and database handling using perl.
6. To explore basics of two way communication between client and server using python and perl

Prerequisites: Knowledge of some programming language like C, Java

Content:

Sr. No	Module Name	Detailed Content
1	Python basics	Data types in python ,Operators in python, Input and Output, Control statement, Arrays in python, String and Character in python, Functions, List and Tuples, Dictionaries Exception, Introduction to OOP, Classes , Objects , Interfaces, Inheritance
2	Advanced Python	Files in Python, Directories, Building Modules, Packages, Text Processing, Regular expression in python.
3	Data Structure in Python	Link List, Stack, Queues, Dequeues
4	Python Integration Primer	Graphical User interface ,Networking in Python , Python database connectivity, Introduction to Django
5	Basics of Perl	Perl Overview, Variables, Control Statements, Subroutines, Objects, Packages and Modules
6	Perl advanced	Working with Files, Data manipulation, Database Systems, Networking

Text Books

1. Core Python Programming, Dr. R. Nageswara Rao, Dreamtech Press
2. Beginning Python: Using Python 2.6 and Python 3.1. James Payne, Wrox publication
3. Perl: The Complete Reference. Second Edition. Martin C. Brown, McGraw-Hill
4. Introduction to computing and problem solving using python , E Balagurusamy, McGraw Hill Education

Reference Book

1. Perl Black Book, 2nd Edition: Steven Holzner, Dreamtech Press
2. Learn Python the Hard Way: (3rd Edition) (Zed Shaw's Hard Way Series)
3. Python Projects , Laura Cassell, Alan Gauld, wrox publication

Digital Material:

1. "The Python Tutorial", <http://docs.python.org/release/3.0.1/tutorial/>
2. Beginning Perl, <https://www.perl.org/books/beginning-perl/>
3. <http://spoken-tutorial.org>
4. www.staredusolutions.org

Suggested experiments using Python:

1. Exploring basics of python like data types (strings, list, array, dictionaries, set, tuples) and control statements.
2. Creating functions, classes and objects using python. Demonstrate exception handling and inheritance.
3. Exploring Files and directories
 - a. Python program to append data to existing file and then display the entire file
 - b. Python program to count number of lines, words and characters in a file.
 - c. Python program to display file available in current directory
4. Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes.
5. Menu driven program for data structure using built in function for link list, stack and queues.
6. Program to demonstrate CRUD(**create, read, update and delete**) operations on database (SQLite/MySQL) using python.
7. Creation of simple socket for basic information exchange between server and client.
8. Creating web application using Django web framework to demonstrate functionality of user login and registration (also validating user detail using regular expression).

Suggested experiments using Perl:

10. Exploring various data type , loops and conditional statement in perl. And Creating functions, packages and modules in perl.
11. Program to demonstrate use of objects and classes in perl.
12. Program to demonstrate file handling, data manipulation and use of regular expression for text processing in perl
13. Program to send email and read content of URL.

Term Work:

Students will submit term work in the form of journal that will include:

1. At least 12-14 programs.
2. One mini-project in a group 2-3 student.
3. Two assignments covering whole syllabus.

Term Work (25) = 15 marks (Experiments & Assignments)

+ 10 marks (Mini Project)

+ 05 marks (Attendance)

Practical and oral examination will be based on suggested practical list and entire syllabus.