College of Engineering, Pune

(An Autonomous Institute of Govt. of Maharashtra, Permanently Affiliated to S.P. Pune University)

Department of Computer Engineering

Curriculum Structure & Detailed Syllabus (UG Program)

Second Year B. Tech.

(Effective from: A.Y. 2020-21)

B.Tech. Computer Engineering

Programme Educational Objectives (PEOs):

I. To create graduates with sufficient capabilities in computer engineering who can become researchers, entrepreneurs and software professionals to satisfy the needs of the core industry, research, academia and society at large.

II. To build ability to continuously learn the latest trends in computer engineering and engage in lifelong learning process.

III. To build engineers aware of professional ethics of the software Industry, and equipped with basic soft skills essential for working in community and professional teams.

Programme Outcomes (POs):

At the end of the program, the graduates will

1. Computer engineering knowledge: Apply the knowledge of mathematics, science, computer engineering fundamentals, and emerging fields of computer engineering to the solution of complex real life problems.

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

3. Design/development of solutions: Design solutions for complex computer engineering problems and design system components or processes that meet the specified needs considering public health and safety, and the cultural, societal, and environmental considerations.

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

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern computer engineering and IT tools including FOSS tools.

6. Social responsibility: Apply reasoning informed by the contextual knowledge to assess social, health, safety, legal and cultural issues and the consequent responsibilities.

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

8. Ethics: Demonstrate knowledge and practice of engineering ethics.

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

10. Communication: Communicate effectively with engineering community and with society at large, demonstrating ability to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the computer engineering, finance and management principles.

12. Life-long learning: Recognize the need for, and ability to engage in independent and life-long learning.

Programme Specific Outcomes (PSOs)

Students will be able to

1. Demonstrate competence in Programming Technologies.

2. Design, implement, test software solutions in core Computer Engineering areas including Computer Networks, Databases, Systems Software, Computer Architecture, Artificial Intelligence, Software Engineering

3. Acquire and demonstrate skills in emerging area like Information Security, Data Science, Natural Language Processing, Cloud Computing, etc.

Correlation between the PEOs and the POs

PO→ PEO↓	a	b	c	d	e	f	g	h
I	✓	✓	✓		✓	✓	✓	✓
II	✓	✓	✓	✓	✓	✓	✓	✓
III		✓	✓	✓	✓			

Correlation between the PEOs and the PSOs

PSO→ PEO↓	1	2	3
I	✓	✓	
II	✓	✓	
III	✓	✓	

List of Abbreviations

Sr. No.	Abbreviation	Stands for:
1	BSC	Basic Science Course
2	HSMC	Humanities, Social Sciences and Management Course
3	IFC	Program Core Course
4	LC	Laboratory Course
5	MLC	Mandatory Learning Course
6	PCC	Programme Core Course
7	SBC	Skill Based Course

CURRICULUM STRUCTURE OF S. Y. B. TECH (Computer Engineering)

Effective from A. Y. 2020-2021

Semester III (Structure for Regular Students)

Sr. No.	Course Code	Course Type	Course Name	Teaching Scheme			Credits
				L	T	P	
1	CT-20001	BSC	Ordinary Differential Equations and Multivariate Calculus	2	1	0	3
2	ML-20002	MLC	Professional Laws, Ethics, Values and Harmony	1	0	0	0
3	HS-20001	HSMC	Innovation and Creativity	1	0	0	1
4	CT-20002	SBC	Development Tools Laboratory	1	0	2	2
5	ICE(IF)-20001	IFC	Feedback Control Systems	1	1	0	2
6	CT-20003	PCC	Data Structures and Algorithms – I	2	0	0	2
7	CT-20004	LC	Data Structures and Algorithms -I Laboratory	0	0	2	1
8	CT-20005	PCC	Digital Logic Design	3	0	0	3
9	CT-20006	LC	Digital Logic Design Laboratory	0	0	2	1
10	CT-20007	PCC	Discrete Structures and Graph Theory	2	1	0	3
11	CT-20008	PCC	Principles of Programming Languages	3	0	0	3
12	CT-20009	LC	Principles of Programming Languages Laboratory	0	0	2	1
				16	3	8	**22**
			Total	27			

Semester III (Structure for Lateral Entry Students)

Sr. No.	Course Code	Course Type	Course Name	Teaching Scheme			Credits
				L	T	P	
1	MA-16003	BSC	Linear Algebra and Univariate Calculus	4	1	0	5
2	ML-20002	MLC	Professional Laws, Ethics, Values and Harmony	1	0	0	0
3	HS-20001	HSMC	Innovation and Creativity	1	0	0	1
4	CT-20002	SBC	Development Tools Laboratory	1	0	2	2
5	ICE(IF)-20001	IFC	Feedback Control Systems	1	1	0	2
6	CT-20003	PCC	Data Structures and Algorithms – I	2	0	0	2
7	CT-20004	LC	Data Structures and Algorithms - I Laboratory	0	0	2	1
8	CT-20005	PCC	Digital Logic Design	3	0	0	3
9	CT-20006	LC	Digital Logic Design Laboratory	0	0	2	1
10	CT-20007	PCC	Discrete Structures and Graph Theory	2	1	0	3
11	CT-20008	PCC	Principles of Programming Languages	3	0	0	3
12	CT-20009	LC	Principles of Programming Languages Laboratory	0	0	2	1
13	PH-16001	BSC	Foundation of Physics	3	0	0	3
				21	3	8	**27**
			Total	32			

Semester IV (Structure for Regular Students)

Sr. No.	Course Code	Course Type	Course Name	Teaching Scheme			Credits
				L	T	P	
1	MA-20004	BSC	Vector Calculus and Partial Differential Equations	2	1	0	3
2	AS-20001	BSC	Biology for Engineers	3	0	0	3
3	CT-20018	SBC	Rapid Prototyping Practice Using Object Oriented Programming	1	0	2	2
4	ICE(IF)-20002	IFC	Sensors and Automation	1	0	2	2
5	CT-20012	PCC	Theory of Computation	3	1	0	4
6	CT-20013	PCC	Microprocessor Techniques	3	0	0	3
7	CT-20016	LC	Microprocessor Techniques Laboratory	0	0	2	1
8	CT-20014	PCC	Data Structures and Algorithms – II	2	0	0	2
9	CT-20017	LC	Data Structures and Algorithms - II Laboratory	0	0	2	1
10	CT-20015	PCC	Data Communication	3	0	0	3
				18	2	8	**24**
			Total	28			

Semester IV (Structure for Lateral Entry Students)

Sr. No.	Course Code	Course Type	Course Name	Teaching Scheme			Credits
				L	T	P	
1	MA-20005	BSC	Multivariate Calculus and Differential Equations*	4	1	0	5
2	AS-20001	BSC	Biology for Engineers	3	0	0	3
3	CT-20018	SBC	Rapid Prototyping Practice Using Object Oriented Programming	1	0	2	2
4	ICE(IF)-20002	IFC	Sensors and Automation	1	0	2	2
5	CT-20012	PCC	Theory of Computation	3	1	0	4
6	CT-20013	PCC	Microprocessor Techniques	3	0	0	3
7	CT-20016	LC	Microprocessor Techniques Laboratory	0	0	2	1
8	CT-20014	PCC	Data Structures and Algorithms – II	2	0	0	2
9	CT-20017	LC	Data Structures and Algorithms -II Laboratory	0	0	2	1
10	CT-20015	PCC	Data Communication	3	0	0	3
				20	2	8	**26**
			Total	30			

(CT-20001) Ordinary Differential Equations and Multivariate Calculus

S.Y. B. Tech. Semester III (All Branches)

Teaching Scheme	**Examination Scheme**
Lectures: 2 Hrs / Week	Internal Test 1: 20 marks
Tutorials : 1 hr / week	Internal Test 2: 20 marks
	End Sem. Exam: 60 marks

Course Outcomes:

Students will be able to

1. **Identify** first order ordinary differential equations, tell Laplace transform formulae, define functions of several variables.
2. **Understand** basic concepts of higher order ordinary differential equations, level curves and level surfaces.
3. **Solve** linear differential equations using different methods, find Laplace transforms of functions using properties and theorems, evaluate directional derivatives and extreme values.
4. **Prove** theorems, solve ordinary differential equations using Laplace transforms, identify orthogonal trajectories, optimize functions subject to given constraints.
5. **Apply** concepts of ordinary differential equations and multivariate calculus to various applications including real life problems.

Unit I : Review of first order differential equations, Reduction of order, linear differential equations, homogeneous higher order linear differential equations, non-homogeneous higher order linear differential equations with constant coefficients and reducible to differential equations with constant coefficients (method of undetermined coefficients and method of variation of parameters), systems of differential equations, applications to orthogonal trajectories, mass spring systems and electrical circuits.

[11 Hrs]

Unit II: Laplace Transforms, its properties, Unit step function, Dirac delta functions, Convolution Theorem, periodic functions, solving differential equations using Laplace transform.

[08 Hrs]

Unit III: Functions of several variables, level curves and level surfaces, partial and directional derivatives, differentiability, chain rule, local extreme values and saddle points, constrained optimization.

[07 Hrs]

Text Books:

- Thomas’ Calculus (14th edition) by Maurice D. Weir, Joel Hass, Frank R. Giordano, Pearson Education.
- Advanced Engineering Mathematics (10th edition) by Erwin Kreyszig, Wiley eastern Ltd.

Reference Books:

- Calculus for Scientists and Engineers by K.D Joshi, CRC Press.
- A Course in Multivariate Calculus and Analysis by Sudhir Ghorpade and Balmohan Limaye, Springer Science and Business Media.
- Differential Equations with Applications and Historical notes by George Simmons, Tata Mc-Graw Hill publishing company Ltd, New Delhi.
- Advanced Engineering Mathematics by C.R. Wylie, McGraw Hill Publications, New Delhi.
- Advanced Engineering Mathematics (7th edition) by Peter V. O' Neil, Thomson. Brooks / Cole, Singapore.
- Advanced Engineering Mathematics (2nd edition) by Michael D. Greenberg, Pearson Education.
- Advanced Engineering Mathematics by Chandrika Prasad and Reena Garg, Khanna Publishing Company Private Limited, New Delhi.

Note 1:

- To measure CO1, questions may be of the type- define, identify, state, match, list, name etc.
- To measure CO2, questions may be of the type- explain, describe, illustrate, evaluate, give examples, compute etc.
- To measure CO3, questions will be based on applications of core concepts.
- To measure CO4, questions may be of the type- true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.
- To measure CO5, some questions may be based on self-study topics and also comprehension of unseen passages.

Note 2 :

All the Course outcomes 1 to 3 will be judged by 75% of the questions and outcomes 4 and 5 will be judged by 25 % of questions.

(MA-16003) Linear Algebra and Univariate Calculus

S.Y. B. Tech. (Lateral Entry Students) Sem III

Teaching Scheme	**Examination Scheme**
Lectures: 4 Hrs / Week	Internal Test 1: 20 marks
Tutorials : 1 hr / week	Internal Test 2: 20 marks
	End Sem. Exam: 60 marks

Objectives

Basic necessity for the foundation of Engineering and Technology being mathematics, the main aim is, to teach mathematical methodologies and models, develop mathematical skills and enhance thinking power of students.

Course Outcomes:

Students will be able to

1. know and recall core knowledge of the syllabus. (To measure this outcome, questions may be of the type- define, identify, state, match, list, name etc.)
2. understand basic concepts. (To measure this outcome, questions may be of the type- explain, describe, illustrate, evaluate, give examples, compute etc.)
3. analyze the problem and apply the appropriate concept. (To measure this outcome, questions will be based on applications of core concepts)
4. give reasoning. (To measure this outcome, questions may be of the type- true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.)
5. apply core concepts to new situations. (To measure this outcome, some questions will be based on self-study topics and also comprehension of unseen passages.)

Unit I : Matrices and linear equations: basic properties of matrices, row operations and Gauss elimination, Determinants and their basic properties. Basic concepts in linear algebra: vector spaces, subspaces, linear independence and dependence of vectors, bases, dimensions. Rank of a matrix . Applications to systems of linear equations.

[15 Hrs]

Unit II : Rank-nullity theorem, Eigen values, Eigen vectors and their basic properties, diagonalization.

[12 Hrs]

Unit III : Review of limits, continuity and differentiability, Mean value theorems, Taylor's theorem, local extrema, increasing and decreasing functions, concavity, points of inflection.

[12 Hrs]

Unit IV : Surface area, integrals by special techniques: reduction formulae, arc length, solids of revolution, improper integrals, tests for convergence, Gamma and Beta functions.

[13 Hrs]

Text Books :

- Thomas' Calculus (14th edition) by Maurice D. Weir, Joel Hass, Frank R. Giordano, Pearson Education.
- Advanced Engineering Mathematics (10th edition) by Erwin Kreyszig, Wiley eastern Ltd.

Reference Books :

- Introduction to Linear Algebra (2nd edition) by Serge Lang, Springer.
- Elementary Linear Algebra (10th edition) by Howard Anton and Chris Rorres, John Wiley and sons.
- Calculus for Scientists and Engineers by K.D Joshi, CRC Press.
- A Course in Calculus and Real Analysis (1st edition) by Sudhir Ghorpade and Balmohan Limaye, Springer-Verlag, New York.
- Advanced Engineering Mathematics by C.R. Wylie, McGraw Hill Publications, New Delhi.

- Advanced Engineering Mathematics (7th edition) by Peter V. O' Neil, Thomson. Brooks / Cole, Singapore.
- Differential Calculus by Shanti Narayan, S. Chand and company, New Delhi.
- Applied Mathematics Vol. I (Reprint July 2014) by P.N. Wartikar and J.N. Wartikar, Pune Vidyarthi Griha Prakashan Pune.
- Advanced Engineering Mathematics by Chandrika Prasad and Reena Garg, Khanna Publishing Company Private Limited, New Delhi.

Note 1:

- All the Course outcomes 1 to 3 will be judged by 75% of the questions and outcomes 4 and 5 will be judged by 25 % of questions.

(ML-20002) Professional Laws, Ethics, Values and Harmony

Teaching Scheme	**Examination Scheme**
	Credits: 0
Lectures: 1 Hr / Week	**Total -** 100 Marks Continuous evaluation- Assignments / Presentations/Test

Course Outcomes :

Students will be able to

1. grasp the meaning of the concept - Law
2. get an overview of the laws relating to Engineers
3. apprehend the importance of being a law abiding person
4. self-explore by using different techniques to live in harmony at various levels
5. analyze themselves and understand their position with respect to the moral
6. and ethical character needed for a successful and satisfactory work life

Unit I: Concept of Law : Understanding Essentials of a Valid Contract and he basics of contract law protecting rights and obligations

[02 Hrs]

Unit II: Law of Torts

Introduction to the Law of Torts and the basics to protect oneself and the company
Law affecting the Workplace
Employers Responsibilities / Duties
Hiring Practices
Introduction to Intellectual Property Law

[03 Hrs]

Unit III: Professional Code of Conduct for Engineers Relationship between Law and Ethics.

[01 Hrs]

Unit IV: Self Awareness Understanding oneself and others; Johari Window- Concept, explanation, implementation

[02 Hrs]

Unit V: Needs & Self

Needs and its importance; Understanding harmony and its relevance in actualization at personal and professional levels .

[02 Hrs]

Unit VI: Ethics and values

Professional ethics and their importance for students; Understanding the importance of values & their application in everyday life

[02 Hrs]

Reference Books :

- Business Law- By Saroj Kumar
- Law of Contract- By Avtar Singh
- Business Law- By G K Kapoor
- Business & Commercial Laws – By Sen & Mitra
- Business Law for Engineers- by Calvin Frank Allen
- Hilgard, E. R.; Atkinson, R. C. & Atkinson, R.L. (1975). *Introduction to Psychology*. 6th Edition. New Delhi: Oxford and IBH Publishing Co. Pvt. Ltd.
- Govindarajan, M; Natarajan, G. M. & Senthilkumar, V.S. (2013). *Professional Ethics & Human Values*. Prentice Hall: New Delhi
- Gogate, S. B. (2011). Human Values & Professional Ethics. Vikas Publishing: New Delhi.
- Govindarajan, M; Natarajan, G. M. & Senthilkumar, V.S. (2013). Professional Ethics & Human Values. Prentice Hall: New Delhi
- Jayshree Suresh, Raghavan B.S.(2016). Human Values & Professional Ethics: S Chand & Company.Pvt.Ltd: New Delhi.

(HS-20001) Innovation and Creativity

Teaching Scheme	**Examination Scheme**
Lectures: 1 Hr/week	"To be declared by the Instructor"

Course Outcomes

At the end of the course, students will demonstrate the ability to:

1. Understand to creativity and innovation terminologies
2. Explore personal and organizational roadblocks in participating in the creative process
3. Apply practical tips to discover the innovative /creative potential within the human being.

4. Study frameworks, strategies, techniques for conceiving ideas.
5. Develop new ways of thinking and Learn the entire innovation cycle.
6. Understand different ways to protect innovation, basics on Patents and process
7. Apply techniques learnt in the course to articulate, refine and pitch a new product or service project

Course Contents

Introduction to concepts of creativity / invention / innovation and their importance in present knowledge world. Components of the creative process, Analogy/model to represent the creative process.

Understanding persons' Creative potential. Blockages in practicing creative process – Mindset and belief systems. Myths and misconceptions about creativity.

Practical Tips to discover and apply one's creative potential, remove blockages, deal with external factors. Importance of synergistically working in a team. Harnessing creativity from nature.

Idea conception,Idea Brainstorming sessions, Idea Evaluation, Protection/Patent review,Principles of innovation, Review of systematic strategies and methods for innovation,Innovation case study,Review of Idea/Prototype /Product and Market Plan.

Applications Exercise / Assignment: at the end of the course, the student will create teams, presents their innovative ideas, and applies their learning in practice.

Reference Books:

- Paul B. Paulus, Bernard A. Nijstad, The Oxford Handbook of Group Creativity and Innovation, Oxford University Press,2019.
- Jeff Dyer, Hal Gregersen, Clayton M. Christensen, " The Innovator's DNA: Mastering the Five Skills of Disruptive Innovators, Harvard Business Review Press, 2011.
- Paddy Miller, Thomas Wedell-Wedellsborg, "Innovation as Usual: How to Help Your People Bring Great Ideas to Life, Harvard Business Review Press, 2013.

(CT-20002) Development Tools Laboratory

Teaching Scheme	**Examination Scheme**
Lectures: 1 Hrs / Week	Practical/Oral Exam: 30 marks
Laboratory : 2 Hrs / Week	Term work: 70 marks

Course Outcomes

At the end of the course, the student will be able to:

1. Develop an application in a group using GIT, demonstrating ability to work remotely, push, and pull.
2. Write a report in a specified format using LaTEX.
3. Demonstrate programming ability using Unix Shell.

Course Contents

LaTEX: Basic syntax, compiling and creating documents; Document structure, sections, paragraphs; packages, Math, Adding Images, Drawing images (using tools like Inkscape) Table of contents; Source code, graphs (using tools like Graphviz), Adding references, different templates, IEEE format, Bibliography

[4 Hrs]

Shell Programming: Introduction to Linux commands, concept of shell, shell variables, getcwd() and pwd; Introduction to shell programming features: Variables declaration & scope, test, return value of a program, if-else and useful examples, for and while loop, switch case; Shell functions, pipe and redirection, wildcards, escape characters; Awk script: Environment and workflow, syntax, variables, operators, regular expressions, arrays, control flows, loops, functions, output redirections

[6 Hrs]

GIT: Creating a project using git locally, add, commit, status, diff; branch and merge, GIT: cloning a remote repo, working with a remote repo – git push, pull, fetch; creating issues and pull requests; working on a project in a distributed fashion

[4 Hrs]

Suggested List of Assignments

1. Format a given essay using sections, paragraphs, headings in LaTEX.
2. Format a given report in IEEE format using LaTEX.
3. Write a shell program which reads a set of unspecified count of numbers and prints their sum and average.
4. Write a shell program to extract a compressed file in any format (zip, tar.gz, tar.gz2, .tar, .bz2, .gz, .rar, .Z, .7z, etc)
5. Write a shell program to convert a CSV file of contacts, into a VCF file.
6. Write a shell program to sort all files stored in a given folder hierarchy, on their size.
7. Write a shell script to manage a todo list from command line. The script should be able to add, remove, list, sort, prepend, append, deduplicate todo-items
8. Write a program that scans a file line by line, splits each input line into fields later, compares input line/fields to pattern and performs action(s) on matched lines
9. Develop a program using git locally. E.g. add the exponent operator to the calculator program that you wrote. Demonstrate the ability to do git add, commit, status, diff.
10. Create a branch in your calculator program to do hexadecimal calculation. Write code and develop two branches. Merge the two to have a decimal/hex calculator. Demonstrate git branch, merge capability.
11. In a group of 3, create a github/gitlab repo. Raise issues, send pull requests, do the local and remote merges and finally get a synced local repo. For example, in the calculator project one student to become the developer, the other two to create issues and send pull requests for features like adding an operator, developing pulling those requests. Rotate the roles and repeat.

References

LaTEX

- Leslie Lamport, "LaTeX: A document preparation system", User's guide and reference manual, 2nd Edition, 1994, by Addison-Wesley Professional. ISBN 0201529831, 9780201529838
- Stefan Kottwitz, "LaTeX Beginner's Guide: Create High-quality and Professional-looking Texts, Articles, and Books for Business and Science Using LaTeX, Packt Publishing, 2011. ISBN: 1847199860, 9781847199867
  "https: //www.latex-project.org/
- Introduction to LaTEX, MIT
  http: //web.mit.edu/rsi/www/pdfs/new-latex.pdf
- A simple guide to LaTeX - Step by Step
  https: //www.latex-tutorial.com/tutorials/

Shell

- Bash Guide for Beginners: https://www.tldp.org/LDP/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf
- Bash Reference Manual https://www.gnu.org/software/bash/manual/bash.pdf
- Tutorials on Shell Programming https://www.shellscript.sh/
  https://www.tutorialspoint.com/unix/shell_scripting.htm

GIT

- Pro GIT Book https: //github.com/progit/progit2/releases/download/2.1.204/progit.pdf

(ICE(IF)-20001) Feedback Control System
(Interdisciplinary Foundation Course – 1)

Teaching Scheme	**Examination Scheme**
Lectures: 1 Hr / Week	**Test 1:** 20 Marks:
Tutorials: 1 Hr / Week	**Test 2:** 20 Marks:
	End-Sem Exam: 60 marks:

Pre-requisite

Basics of Signals and System, Basic Physics laws, Laplace Transform.

Course Outcomes

Students will be able to

1. Understand the effects of feedback in linear systems.
2. Develop a mathematical model of electrical, mechanical and electromechanical system.
3. Ability to apply various feedback control analysis and design methods.
4. Design a closed-loop system to meet a desired behavior

Course Contents

Unit I: Importance of control systems, Control situations in Industry and around, classification of control system, transfers function, System modeling in the time domain.

[6hrs]

Unit II: Dynamics of electrical and mechanical systems. Signal flow graph, System response in the time domain, Time-domain specifications. Time response analysis, 1st, 2nd and higher order systems. Basic properties of feedback, Advantage of feedback, Steady state errors (SSE) for feedback systems, static error constants and system types.

[8 hrs]

Unit III: Stability of open loop and closed loop systems, Routh-Hurwitz stability criterion. Root locus techniques, Root locus construction rules, Effects of Pole and Zeros. Using root-locus ideas to design controller, reducing steady-state error.

[6 hrs]

Tutorials:

1) To study SISO and MIMO Systems
2) To find the transfer function of unknown system (electrical / mechanical / Electromechanical system)
3) Develop a MATLAB/Simulink program to generator standard test signals.
4) Analysis of second order (R-L-C) system in time domain.
5) Write a program to find step and ramp response of a second order system and verify with physical system.
6) Develop a Simulink model to find steady state error for a type 0, type 1 and type 2 systems.
7) Write a program to find Routh table and comment on its stability
8) Write a program to design controller using root locus technique

Text Books

- Norman Nise, Control System Engineering, Wiley International, sixth edition, 2011.
- Nagrath and Gopal, Control System Engineering-, New Age International Publication, fifth edition,2003

Reference Books:

- G. Goodwin, S. Graebe, Mario Salgado, Control System Design –, Pearson Education, first edition, 2000.
- G. Franklin, J. Powell, A. Naeini, Feedback Control of Dynamic Systems- Pearson, seventh edition, 2014.
- K. Ogata, Modern Control Engineering- Prentice Hall Publications, fifth edition, 2012

Web References:

- NPTEL video lectures "Control Engineering" by M. Gopal, IIT Delhi.

(CT-20003) Data Structures and Algorithms - I

Teaching Scheme	**Examination Scheme**
Lectures: 2 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Write neat code by following coding standards, by selecting appropriate data structure and demonstrate a working solution for a given problem.
2. Think of all possible inputs to an application and handle all possible errors properly.
3. Analyze different possible solutions to a program and select the most efficient one.
4. Write an application requiring an effort of at least 500 lines of code to demonstrate a good working solution.
5. Demonstrate the ability to write reusable code and abstract data types.

Course Contents

Introduction to imperative programming. Syntax of a language for imperative programming. Input-output statements, data, data types, variables, constants, limitations of data types, type conversion, operators, control statements,compilation and execution as independent steps. Functions, Macros, Preprocessor directives, global, static, local variables, arrays and structures, pointers, pointer arithmetic, Dangling pointers and garbage memory, user defined types. writing code in multiple files, Coding standards

[6 Hrs]

Introduction to Data representation and files: Text and binary files, use of various libraries for handling files. Implementation of utilities like 'cat', 'cp', 'mv', etc

[3 Hrs]

Lists: Concept of Abstract Data types (ADT), List as ADT,Features of a Linear data structure, Dynamic memory allocation. self-referential structures, Concept of linked organization of data. Various implementations of List ADT as singly linked list, doubly linked list, circular linked list. Operations on linked lists: insert, delete, traverse, search etc. Applications of linked list: Representation & manipulations of polynomials/sets using linked list concept.

[5 Hrs]

Stacks and Queues: Stack and queue as ADT. Operations on stack and queue. Implementations using arrays and dynamic memory allocation. Application of stack for expression evaluation, expression conversion. Implementation of stack using queue and vice-versa. Recursion and stacks. Problems like maze and knight's tour.

[6Hrs]

Time Complexity Analysis. Characteristics of an algorithm. Analyzing programs. Frequency count. Time and space complexity. Big 'O' 'Ω, Θ, notation. Best, average and worst cases.

[2 Hrs]

Searching, Sorting Algorithms: Searching: linear and binary search algorithm. Searching using key-value in a sequence of records. Insertion, bubble, selection sort algorithms. Sort algorithms on a sequence of records using specified key.Comparative analysis of various searching and sorting algorithms.

[4 Hrs]

Applications: Design of data structures, modules, functions and algorithms for applications like a general-purpose random precision calculator like "bc", 'sort' utility with external sort, 'diff', 'grep', 'head', 'tail', 'dd', 'uniq', 'wc' etc.

[4 Hrs]

Text Books

- "Fundamentals of Data Structures in C", E. Horowitz, S. Sahni, S. Anderson-freed, Second Edition, 2008, University Press, ISBN 978-81-7371-605-8
- "The C Programming Language", B. Kernighan, D. Ritchie, Second Edition, 2015, Pearson Education India; ISBN 81-203-0596-5

Reference Books:

- "Data Structures using C", Y. Langsam, M. Augenstin and A. Tannenbaum, First Edition, 2002, Pearson Education Asia, ISBN 978-81-317-0229-1
- "C++: The Complete Reference", Herbert Schildt, Fourth Edition, 2002, The McGraw-Hill company, ISBN 0-07-222680-3
- "Fundamentals of Data Structures in C++", Ellis Horowitz, S. Sahni, D. Mehta, 2nd Edition, 2008, University Press, ISBN-10: 8173716064
- "An introduction to data structures with Applications", Jean-Paul Tremblay, Paul. G. Soresan, 2nd Edition, 1984, Tata Mc-Graw Hill International Editions, ISBN-0-07-462471-7

(CT-20004) Data Structures and Algorithms –I Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2 Hrs / Week	Continuous evaluation: 50 Marks
	Mini Project: 25 marks
	End Semester Exam: 25 Marks

Course Outcomes

Students will be able to:

1. Write neat code by following coding standards, by selecting appropriate data structure and demonstrate a working solution for a given problem.
2. Think of all possible inputs to an application and handle all possible errors properly.
3. Analyze different possible solutions to a program and select the most efficient one.

4. Write an application requiring an effort of at least 500 lines of code to demonstrate a good working solution.
5. Demonstrate the ability to write reusable code and abstract data types.

Suggested List of Assignments

1. *Assignments for practice of language syntax, data types and programming constructs*
   a. Write a program to print a histogram of the frequencies of different characters in its input.
   b. Write a program to print sum of digits of a number accepted from user. Example: if the input number is 7654, output should be 7 + 6 + 5 + 4 = 22.
   c. Write a program to determine a string is a palindrome or not. A palindrome is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as madam, racecar
   d. Write a program in C to define a structure for a Customer bank account that holds information like Account Number, Name of account holder, balance, internet banking facility availed(Yes or No), Pin code (422001 to 422013), Account type(saving, recurring, deposit).
      i. Read account details for N customers
      ii. Classify the customer as Golden, Silver and General based on the following criteria: Golden customers: Balance> 10, 00000, Silver Customers: Balance >500000 and <500000, General customers: Balance <500000
      iii. Display the list of customers availing the Internet banking facility.
      iv. Display the customers belonging to a particular geographical location depending on postal code.
      v. Display the customer list as per their account type
   e. Write a function squeeze (s, c), which removes all occurrences of the character c from the string s.
   f. Write a C program to accept a string and change the case of each character of the string. Example “ THIs Is a C Program” changes to “thiSiS A c pROGRAM”
   g. Write a C program to swap two numbers using a function.
2. Draw a diagram of data structures created by *given code* using a tool like xfig.
3. Write a program to compute x^y based on using base-3 presentation of a number. In the program, write a function which computes x^y
4. Write a program to remove duplicate doubles from an array of doubles. In the program, write a function which accepts an array of doubles and removes the duplicates from the array and has return type void.
5. Compare the time complexity of two sorting algorithms, by following the given steps. Create a set of data files with count of integers varying into thousands and millions. Sort the files using both the algorithms. Plot graph of the time taken by both the programs using tool like *gnuplot*. Compare the graphs and comment on the time complexity theoretically predicted and practically observed.
6. Write a function which evaluates an infix expression, without converting it to postfix. The input string can have spaces, (,) and precedence of operators should be handled.
7. Implement a queue (that is write queue.c and queue.h only) of characters, such that on an enqueue, the char is added at the end of queue, and on a dequeue the first element is taken out, but the queue uses only a 'head' pointer and not a 'tail pointer.

8. Write an data type called "Integer". The data type should represent integers of unlimited length.
9. Implement a data type to represent a Polynomial with the operations like create an empty polynomial, insert an entry into polynomial, add two polynomials and return the result as a polynomial, evaluate a polynomial, etc.
10. Implement a Set data type using sequentially linked structures. Support operations like create an empty set, insert an element into set, do a union of two sets and return results as a set, etc.
11. Write a sorting program with the following features: Reads data from a text file and sorts it alphabetically by default. If the file has data in rows and columns (separated by space or tab) then allows sorting on a particular column. Allows any sort using numeric or alphbetical ordering.
12. **Mini-project**: Write an application demonstrating your skills in defining a problem, writing down the requirements carefully, designing a modular solution with clear separation of abstract data types and their use, design of proper function prototypes and division of work among functions. The application can be a unix command re-implemented e.g grep, sort, diff, calculator etc. or a task defined after discussion with the instructor.

(CT-20005) Digital Logic Design

Teaching Scheme	**Examination Scheme:**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Apply the knowledge of number systems and codes in problem solving related to code conversion and number system.
2. Optimize circuit design.
3. Explain the fundamental concepts of combinational logic devices and design them.
4. Explain fundamentals of sequential logic devices and design them.
5. Analyze and design algorithmic state machines.

Course Contents

Number systems and codes : Binary number systems, Octal number system, hexadecimal number system, Signed binary numbers, Binary arithmetic, 1's and 2's complement, Introduction to gates, Review of Boolean algebra and DeMorgan's law, Minimization of Boolean function using Karnaugh Map (up to four variable), SOP-POS, Quine - Mclusky methods, Code conversions- Binary code to gray code and gray to binary, BCD to Excess – 3, Excess – 3 to BCD code etc.

[8 Hrs]

Combinational Logic Circuits: Integer adders/subtractors, Ripple carry adder and Carry look ahead adder, Integer subtractions using adders, Design of Combinational Circuits using Programmable Logic Devices (PLDs): Programmable Read Only Memories (PROMs), Programmable Logic Arrays (PLAs), Programmable Array Logic (PAL) devices, RAM, ROMs, EPROM, and MOS-static RAM cell.

[8 Hrs]

MSI Combinational Logic Circuits: Modular combinational logic elements, Overview & implementation of multiplexer/ demultiplexer, Implementation of Combinational Logic Circuits using mux / demux, Decoders, Encoders, Priority encoders.

[4 Hrs]

Sequential Logic Circuits: Latches: RS latch and JK latch, Flip-flops-RS, JK, T and D flip flops, Master-slave flip flops, Edge-triggered flip-flops. Analysis and Design of Synchronous Sequential Circuits: Introduction to sequential circuits, Characteristic table, Characteristic equation and Excitation table.

[8 Hrs]

Modular sequential logic circuits: Registers, Design of Synchronous / Asynchronous using different flip-flops. Overview of Shift registers. Counters-Synchronous / Asynchronous, Up-down, Ring, Johnson counter**.**

[6 Hrs]

Algorithmic State Machines: ASM charts, notation, RTL notation and implementation design of simple controller, multiplexer controller method.

[6 Hrs]

Text Books

- "Digital Design" M Morris Mano, 5th Edition, 2013, Pearson Education, ISBN-10: 0-13-277420-8 / ISBN-13: 978-0-13-277420-8.
- "Modern Digital Electronics", R.P. Jain, 4th Edition, 2009, Tata McGraw-Hill, ISBN 10: 0070669112 / ISBN 13: 9780070669116.

Reference Books:

- "Digital Design: Principles and Practices", John F. Wakerly, 5th edition, 2018, Pearson, ISBN-13: 9780134460093.
- "Fundamentals of digital circuits" A. Anand Kumar, 4th edition, 2016, PHI publication, ISBN-10: 8120350529 / ISBN-13: 978 - 8120350526.

(CT-20006) Digital Logic Design Laboratory

Teaching Scheme	**Examination Scheme:**
Laboratory: 2 Hrs / Week	Practical/Oral Exam: 50 marks
	Term Work: 50 marks

Course Outcomes

Students will be able to:

1. Design and analyze combinational logic circuits.
2. Design and analyze sequential logic circuits.

List of Assignments:

1. Verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, & EX-NOR).
2. Simplify Boolean function using K-map method and implement using basic gate.
3. Design and Implement following code conversion
   a. Binary to Gray
   b. Gray to Binary
   c. Excess – 3 code to BCD
   d. BCD to Excess – 3 code
4. Design and verify a half adder and full adder
5. Design and verify a half subtracter and full subtracter.
6. Design a combinational circuit whose output is the 2’s complement of the input number.
7. Design and implement a 4-bit adder/subtractor circuit with ADD/SUB control line using 4 bit adder IC.
8. Implement BCD adder using 4 bit adder IC.
9. Implementation of Multiplexer, Demultiplexer, Encoder and Decoder.
10. Design and verify the operation of RS, D, JK and T flip-flops using logic gates.
11. Design synchronous and asynchronous up down counter.
12. Design Mod counter.
13. Design sequence generator using JK, T and D flop-flop.
14. Implement and verify operation of ring and Johnson counter.
15. Design and verify the various operations of shift registers.

Text Books

- “Digital Design” M Morris Mano, 5th Edition, 2013, Pearson Education, ISBN-10: 0-13-277420-8 / ISBN-13: 978-0-13-277420-8.
- “Modern Digital Electronics”, R.P. Jain, 4th Edition, 2009, Tata McGraw-Hill, ISBN 10: 0070669112 / ISBN 13: 9780070669116.

Reference Books:

- "Digital Design: Principles and Practices", John F. Wakerly, 5th edition, 2018, Pearson, ISBN-13: 9780134460093.
- “Fundamentals of digital circuits” A. Anand Kumar, 4th edition, 2016, PHI publication, ISBN-10: 8120350529 / ISBN-13: 978 - 8120350526.

CT-20007) Discrete Structures and Graph Theory

Teaching Scheme	**Examination Scheme:**
Lectures: 2 Hrs / Week	Assignment/Quizzes : 40 marks
Tutorials: 1 hr / week	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Explain formal logic and different proof techniques.
2. Recognize relation between different entities using sets, functions, and relations.
3. Use Chinese Remainder Theorem & the Euclidean algorithm for modular arithmetic.
4. Solve problems based on graphs, trees and related algorithms.
5. Relate, interpret and apply the concepts to various areas of computer science.

Course Contents

Set Theory, Logic and Proofs : Propositions, Conditional Propositions, Logical Connectivity, Propositional calculus, predicates and Quantifiers, First order logic, Proofs: Proof Techniques, Mathematical Induction, Set, Combination of sets, Finite and Infinite sets, countable and Uncountable sets, Principle of inclusion and exclusion,

[8 Hrs]

Relations, Functions, Recurrence Relations: Definitions, Properties of Binary Relations, Equivalence Relations and partitions, Partial ordering relations and lattices, Chains and Anti chains. Theorem on chain, Warshall's Algorithm & transitive closure, Recurrence relations. Functions: Definition, Domain, Range, Image, etc. Types of functions: Surjection, Injection, Bijection, Inverse, Identity, Composition of Functions, Generating Function

[8 Hrs]

Number Theory: Basics of Modulo Arithmetic, Basic Prime Number Theory, GCD, LCM, Divisibility, Euclid's algorithm, Factorization, Congruences, inverse , multiplicative inverse, Chinese Remainder Theorem

[4 Hrs]

Counting: Basic Counting Techniques (sum, product, subtraction, division, exponent), Pigeonhole and Generalized Pigeonhole Principle with many examples, Permutations and Combinations and numerical problems, Binomial Coefficients Pascal's, Identity and Triangle

[6 Hrs]

Graphs & Trees: Basic terminology, multi graphs and weighted graphs, paths and circuits, shortest path Problems, Euler and Hamiltonian paths and circuits, factors of a graph, planar graph and Kuratowskis graph and theorem, independent sets, connectivity graph coloring. Trees, rooted trees, path length in rooted trees, binary search trees, spanning trees and, theorems on spanning trees, cut sets , circuits, minimum spanning trees, Kruskal’s and Prim’s algorithms for minimum spanning tree.

[8 Hrs]

Algebraic Systems: Algebraic Systems, Groups, Semi Groups, Monoids, Subgroups, Permutation Groups, Codes and Group codes, Isomorphism and Automorphisms, Homomorphism and Normal Subgroups, Ring, Field.

[6 Hrs]

Text Books

- "Discrete Mathematics and Its Applications", Kenneth H. Rosen, 7th Edition, Tata McGraw-Hill, 2017, ISBN: 9780073383095.
- "Elements of Discrete Mathematics", C. L. LIU, 4th Edition, Tata McGraw-Hill, 2017, ISBN-10: 1259006395 ISBN-13: 978125 9006395.

Reference Books:

- "Discrete Mathematical Structures", G. Shanker Rao, 2nd Edition2009, New Age International, ISBN-10: 8122426697, ISBN-13: 9788122426694
- "Discrete Mathematics", Lipschutz, Lipson, 2nd Edition, 1999, Tata McGraw-Hill, ISBN: 007 463710X.
- "Graph Theory", V. K. Balakrishnan, 1st Edition, 2004, Tata McGraw-Hill , ISBN-10: 0-07-058718-3, ISBN-13: 9780070587182.
- "Discrete Mathematical Structures", B. Kolman, R. Busby and S. Ross, 4th Edition, Pearson Education, 2002, ISBN: 8178085569
- "Discrete Mathematical Structures with application to Computer Science", J. Tremblay, R. Manohar, Tata McGraw-Hill, 2002, ISBN: 0070651426
- "Discrete Mathematics", R. K. Bisht, H. S. Dhami, Oxford University Press, ISBN: 9780199452798

(CT-20008) Principles of Programming Languages

Teaching Scheme	**Examination Scheme:**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Explain, compare and discuss different language translation mechanisms.
2. Explain fundamental concepts of different programming paradigms.
3. Discuss and analyze factors that impact implementation of different programming language concepts and tradeoffs involved.
4. Illustrate different calling conventions.
5. Suggest a suitable programming paradigm for a given problem.

Course Contents

Preliminaries: Reasons for Studying Concepts of Programming Languages. Programming Domains. Language Evaluation Criteria. Influences on Language Design: computer architecture. Language Categories. Language Design Trade-Offs

[5 Hrs]

Implementation Methods : Compilation, Interpretation, Hybrid Implementation, Preprocessors. Programming Environments. Evolution of the Major Programming Languages (Pseudocodes, Assemby, C, C++, FORTRAN, Java). Introduction to assembly language instructions.

[5 Hrs]

Syntax and Semantics Lexical and Syntax Analysis. Names, Bindings, and Scopes, Type checking, Strong Typing. Type Conversions. Short-Circuit Evaluation

[8 Hrs]

Statement-Level Control Structures Subprograms: Introduction. Fundamentals of Subprograms. Design Issues for Subprograms. Local Referencing Environments. Parameter-Passing Methods

[6 Hrs]

Object Oriented Programming: Abstraction and Encapsulation. Structured Data Types. Abstract Data Types. Encapsulation by Subprograms. Type Definitions. Implementation Inheritance. Abstract Data Types. Inheritance. Polymorphism

[6 Hrs]

Exception Handling: Basic Concepts. Design Issues. Exception Handlers . Binding Exceptions to Handlers.
Event Handling: Basic Concepts. Event handling with Programming Language. Comparison of Exception handling & Event Handling. GUI development

[6 Hrs]

Introduction to Logic Programming. Introduction. Applications of Logic Programming
Introduction to Functional Programming. Introduction. Introduction to LISP. Garbage Collection Algorithms. A Comparison of Functional and Imperative Languages.

[6 Hrs]

Text Books

- Scbesta R., "Concepts Of Programming Languages", 4th Edition, Pearson Education, ISBN- 81-7808-161-X
- T. W. Pratt , "Programming Languages", 4th Edition ,Prentice-Hall Of India, ISBN 9780130287199.

Reference Books:

- Ghezzi C, Milano P., Jazayeri M., "Programming Languages Concepts", 3rd Edition, John Wiley and Sons Pvt. Ltd (WSE), ISBN - 0195113063
- Michael L. Scott “Programming Language Pragmatics”, ELSEVIER Publication, ISBN: 81-8147-370-1
- Roosta S., "Foundations of Programming Languages", Thomson, Brooke/Cole, ISBN 981-243-141-1
- Sethi R., "Programming Languages concepts & constructs", 2nd Edition, Pearson Education, ISBN 81 - 7808 - 104 – 0

(CT-20009) Principles of Programming Languages Laboratory

Teaching Scheme	**Examination Scheme:**
Laboratory: 2 Hrs / Week	Practical/Oral Exam: 50 marks
	Term Work: 50 marks

Course Outcomes

Students will be able to:

1. Demonstrate how a stack frame is built using gcc compiler and assembly code.
2. Demonstrate implementation of calling conventions.
3. Demonstrate ability to write simple programs using Object Oriented, Logic and Functional programming concepts.
4. Demonstrate ability to write simple programs using exception handling and event driven programming concepts.

List of Assignments:

1. Compile a C program with *gcc* with various options like -S, --save-temps.
2. Use *objdump* to read the object file of a C program and demonstrate how a stack frame is built and destroyed.
3. Use gdb to read assembly code for C program to find factorial of a number with recursion and demonstrate how parameters are passed using pass by value.
4. Explain how pass by reference works, using assembly code generated using g++ -S on a C++ program.
5. Write a program to develop a software to introduce animals to kids. Implement classes for animals like Dog, Cat, Elephant, Fish etc. With constructor destructor interfaces and variables with public or private access specifies.
6. Write a program to implement various types of geometrical shapes like circle, triangle, square, polygon, and octagon. Circle, triangle, square, polygon, octagon are child classes of base class shape; Square is a polygon, octagon is a polygon.
7. Write a program to demonstrate event and event handlers using Object Oriented Programming.
8. Write a program to raise an exception and handle it.
9. Write a function in LISP to find nth element from a list of m elements.
10. Implement a decision support system for course tracking at a institute using PROLOG. A course is floated by a teacher from a particular department. Various students will enroll a course. Single student can enroll for multiple courses (maximum five per course) provided the student clears the prerequisite courses. A teacher can teach at the most two courses in a semester.

(PH-16001) Foundations of Physics

S.Y. B.Tech. Semester III (Lateral Entry Students)

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs / Week	Internal Test 1: 20 marks Internal Test 2: 20 marks End Sem. Exam: 60 marks

Objectives

Basic necessity for the foundation of Engineering and Technology being mathematics, the main aim is, to teach mathematical methodologies and models, develop mathematical skills and enhance thinking power of students.

Course Outcomes :

Students will be able to

1. Understand classical and wave mechanics to implement for the problems.
2. Understand of the laws of thermodynamics to implement in various thermodynamic systems and processes
3. Understand the basic principles of Electromagnetism and formulate it to solve the engineering problems
4. Aware of limits of classical physics and will be able to use it in the appropriate field in order to solve the problems

Unit I : Oscillations, Waves & Light SHM, characteristics of SHM, Waves, Travelling waves and its equation**,** Types of waves, Principle of Superposition, Stationary waves**,** Light as an EM Wave, graphical representation of EM wave, Interference of light due to thin film (uniform thickness), Antireflection coating, Total Internal reflection, Introduction to Optical fiber and its design..

[07 Hrs]

Unit II : Atomic Nucleus and Nuclear energy : Atomic Nucleus, Nuclear force, Static properties of nucleus, Mass defect and Binding energy, Law of radioactive decay, Half-life, Applications of radioactivity, Nuclear reactions, Q-value of nuclear reaction, Nuclear fission, chain reaction and Nuclear energy.

[07 Hrs]

Unit III : Electrostatics Coulomb's law in vector form ,the electric field, Continuous charge distribution (Line, Surface& Volume)**,** Divergence of E, application of Gauss's law (simple 2 D problems)**,** The curl of E (Faraday's Law), the concept of electric potential V, Potential due to continuous charge distribution.

[07 Hrs]

Unit IV : Magneto statics Steady state current (line current, Surface current and volume current), current densities, Magnetic field due to steady current (Biot-Savart's law), divergence and curl of B, Statement of Ampere's Law (with simple examples).

[07 Hrs]

Unit V : Elements of Thermodynamics Concept of Temperature, Terminology in Thermodynamics, Thermodynamic work, Caparison for Heat and Work, First Law and its applications, Heat engine and Thermal efficiency, Second law, Entropy, Disorder of system, Third law and Principle of Unattainability Absolute Zero (Nernst's Theorem).

[07 Hrs]

Unit VI : Modern physics Drawbacks of Classical Mechanics, Plank's quantum hypothesis, Dual nature of matter, De-Broglie's hypothesis, light as a particle(Compton's experiment), De-Broglie's wavelength, Heisenberg's uncertainty principle(position and momentum), Wave function, its properties, conditions and its physical significance, Free particle solution of wave function..

[07 Hrs]

Reference Books :

- Engineering Physics, Avadhanulu and Kshirsagar.
- Halliday-Resnick (Sixth edition) "Optics", Brij Lal (S. Chand publication)
- Classical Electrodynamics, David Griffith (Pearson India limited)
- H .C. Verma & Halliday-Resnick (Sixth edition), B. B. Laud
- Modern Physics, S. Chand Publication.
- Concepts of Modern Physics, Arthur Beiser, Tata McGraw – Hill Edition

(MA-20004) Vector Calculus and Partial Differential Equations

S.Y. B.Tech. Semester III (All Branches)

Teaching Scheme	**Examination Scheme**
Lectures: 2 Hrs / Week	Internal Test 1: 20 marks
Tutorials : 1 hr / week	Internal Test 2: 20 marks
	End Sem. Exam: 60 marks

Course Outcomes :

Students will be able to

1. Know and recall double / triple integrals, vector differentiation, vector integration, partial differential equations.
2. Understand basic concepts of co-ordinate systems, iterated integrals, gradient, divergence and curl.
3. Evaluate multiple integrals, find area / mass / volume using multiple integrals, evaluate line integrals and surface integrals.
4. Prove theorems, apply Green's / Stoke's / Divergence theorem to different type of problems, model one dimensional heat / wave equations, solve partial differential equations.

5. Apply concepts of vector calculus and partial differential equations to various applications including real life problems.

1. **Know** and **recall** double / triple integrals, vector differentiation, vector integration, partial differential equations.
2. **Understand** basic concepts of co-ordinate systems, iterated integrals, gradient, divergence and curl.
3. **Evaluate** multiple integrals, **find** area / mass / volume using multiple integrals, evaluate line integrals and surface integrals.
4. **Prove** theorems, **apply** Green's / Stoke's / Divergence theorem to different type of problems, **model** one dimensional heat / wave equations, **solve** partial differential equations.
5. **Apply** concepts of vector calculus and partial differential equations to various applications including real life problems.

Unit I : Double integrals in Cartesian and polar co-ordinates, iterated integrals, change of variables, triple integrals in Cartesian, spherical and cylindrical co-ordinates, substitutions in multiple integrals, Applications to Area, Volume, Moments and Center of Mass.

[10 Hrs]

Unit II : Vector differentiation, gradient, divergence and curl, line and surface integrals, path independence, statements and illustrations of theorems of Green, Stokes and Gauss, arc length parameterization, applications.

[07 Hrs]

Unit III : Partial differential equations with separation of variables, boundary value problems: vibrations of a string, heat equation, potential equation, vibrations of circular membranes.

[09 Hrs]

Text Books :

- Thomas' Calculus (14th edition) by Maurice D. Weir, Joel Hass, Frank R. Giordano, Pearson Education.
- Advanced Engineering Mathematics (10th edition) by Erwin Kreyszig, Wiley eastern Ltd

Reference Books :

- Advanced Engineering Mathematics by C.R. Wylie, McGraw Hill Publications, New Delhi.
- Functions of several variables by Wendell Fleming, Springer-Verlag, New York.
- Partial Differential Equations (4th edition) by Fritz John, Springer.
- Advanced Engineering Mathematics (7th edition) by Peter V. O' Neil, Thomson.Brooks / Cole, Singapore.
- Advanced Engineering Mathematics (2nd edition) by Michael D. Greenberg, Pearson Education.
- Advanced Engineering Mathematics by Chandrika Prasad and Reena Garg, Khanna Publishing Company Private Limited, New Delhi.

- Note 1 :

  - To measure CO1, questions may be of the type- define, identify, state, match, list, name etc.
  - To measure CO2, questions may be of the type- explain, describe, illustrate, evaluate, give examples, compute etc.
  - To measure CO3, questions will be based on applications of core concepts.
  - To measure CO4, questions may be of the type- true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.
  - To measure CO5, some questions may be based on self-study topics and also comprehension of unseen passages.

- Note **2 :**
  - All the Course outcomes 1 to 3 will be judged by 75% of the questions and outcomes 4 and 5 will be judged by 25 % of questions.

(MA-20005) Multivariate Calculus and Differential Equations
S.Y. B. Tech. (Lateral Entry Students) Sem III

Teaching Scheme	**Examination Scheme**
Lectures: 4 Hrs / Week	Internal Test 1: 20 marks
Tutorials : 1 hr / week	Internal Test 2: 20 marks
	End Sem. Exam: 60 marks

Objectives

Basic necessity for the foundation of Engineering and Technology being mathematics, the main aim is, to teach mathematical methodologies and models, develop mathematical skills and enhance thinking power of students.

Course Outcomes :

Students will be able to

1. Know first order ordinary differential equations, list Laplace transform formulae, define functions of several variables, double / triple integrals, vector differentiation, vector integration, and partial differential equations.
2. Understand basic concepts of higher order ordinary differential equations, level curves and level surfaces, co-ordinate systems, iterated integrals, gradient, divergence and curl.
3. Solve linear differential equations using different methods, find Laplace transforms of functions using properties and theorems, evaluate directional derivatives and extreme values, evaluate multiple integrals, find area / mass / volume using multiple integrals, evaluate line integrals and surface integrals.

4. Prove theorems, solve ordinary differential equations using Laplace transforms, apply Green’s / Stoke’s / Divergence theorem to different type of problems, model one dimensional heat / wave equations, solve partial differential equations.
5. Apply concepts of multivariate calculus and differential equations to various applications including real life problems.

Unit I : Review of first order differential equations, linear differential equations, homogeneous higher order linear differential equations, non-homogeneous higher order linear differential equations with constant coefficients (method of undetermined coefficients and method of variation of parameters).

[09 Hrs]

Unit II : Laplace Transforms, its properties, Unit step function, Dirac delta functions, Convolution Theorem, periodic functions, solving differential equations using Laplace transform

[07 Hrs]

Unit III : Functions of several variables, level curves and level surfaces, partial and directional derivatives, differentiability, chain rule, local extreme values and saddle points.

[07 Hrs]

Unit IV : Double integrals in Cartesian and polar co-ordinates, iterated integrals, change of variables, triple integrals in Cartesian, spherical and cylindrical co-ordinates, applications to area, mass, and volume.

[12 Hrs]

Unit V : Vector differentiation, gradient, divergence and curl, line and surface integrals, path independence, statements and illustrations of theorems of Green, Stokes and Gauss.

[10 Hrs]

Unit VI : Partial differential equations with separation of variables, boundary value problems: vibrations of a string, one dimensional heat equation

[07 Hrs]

Text Books :

- Thomas’ Calculus (14th edition) by Maurice D. Weir, Joel Hass, Frank R. Giordano, Pearson Education.
- Advanced Engineering Mathematics (10th edition) by Erwin Kreyszig, Wiley eastern Ltd.

Reference Books :

- Calculus for Scientists and Engineers by K.D Joshi, CRC Press.
- A Course in Multivariate Calculus and Analysis by Sudhir Ghorpade and Balmohan Limaye, Springer Science and Business Media.

- Differential Equations with Applications and Historical notes by George Simmons, Tata Mc-Graw Hill publishing company Ltd, New Delhi.
- Functions of several variables by Wendell Fleming, Springer-Verlag, New York.
- Partial Differential Equations (4th edition) by Fritz John, Springer.
- Advanced Engineering Mathematics by C.R. Wylie, McGraw Hill Publications, New Delhi.
- Advanced Engineering Mathematics (7th edition) by Peter V. O' Neil, Thomson.Brooks / Cole, Singapore.
- Advanced Engineering Mathematics (2nd edition) by Michael D. Greenberg, Pearson Education.

  Advanced Engineering Mathematics by Chandrika Prasad and Reena Garg, Khanna Publishing Company Private Limited, New Delhi.

- Note 1 :
  - To measure CO1, questions may be of the type- define, identify, state, match, list, name etc.
  - To measure CO2, questions may be of the type- explain, describe, illustrate, evaluate, give examples, compute etc.
  - To measure CO3, questions will be based on applications of core concepts.
  - To measure CO4, questions may be of the type- true/false with justification, theoretical fill in the blanks, theoretical problems, prove implications or corollaries of theorems, etc.
  - To measure CO5, some questions may be based on self-study topics and also comprehension of unseen passages.

- Note 2 :
  - All the Course outcomes 1 to 3 will be judged by 75% of the questions and outcomes 4 and 5 will be judged by 25 % of questions.

(AS-20001) Biology for Engineers

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs / Week	Internal Test 1: 20 marks
	Internal Test 2: 20 marks
	End Sem. Exam: 60 marks

Course Outcomes :

Students will be able to

1. understand basic biological principles and organizational structure of living systems at molecular level
2. comprehend basic biological principles and organizational structure of living systems at cellular level
3. know Energy transformations and information processing in biological systems
4. appreciate biological process with engineering perspective
5. impart knowledge about the common corridors of biology and engineering and biologically inspired technologies

Unit I : Biomolecules and biopolymers: Structure and Function Organic and inorganic molecules; Unique Properties of water, Vitamins and Minerals, Carbohydrates, Lipids, Amino Acids and proteins, Nucleic Acids (DNA and RNA)

[06 Hrs]

Unit II : Levels of organization of life: Cell as a basic unit of life, prokaryotic and eukaryotic cells, microbes, plant and animal cells; Cell organelles – structure and function; Cell membrane
Levels of organization: cells, tissues, organs, systems & organism.

[06 Hrs]

Unit III : Energy transformations in Chloroplast: Photosynthesis (photochemical & biochemical phase) and ATP generation, Aerobic and anaerobic systems
Energy transformations in Mitochondria: Cellular respiration (glycolysis and Kreb cycle) and ATP generation
Bioenergetics: Thermodynamic principles applied to biology, negative entropy changes in biological systems, Free Energy, Chemical Equilibrium.

[06 Hrs]

Unit IV : Expression and Transmission of Genetic Information: DNA replication, Enzyme driven process of DNA cloning, Protein synthesis- Transcription & translation
Techniques for optimization:
a. At molecular level: Recombinant DNA Technology, DNA hybridization, PCR, DNA microarray.

[06 Hrs]

Unit V : Transport Phenomena in Biological Systems: Membrane channels and ion channels; Fluid flow and mass transfer (nutrients & ions); In plants: Xylem and Phloem; In animals: Blood and Lymph Transport of gases: Oxygen and Carbon dioxide
Heat Transport - Body temperature regulation.
Communication: Cell junctions, Cell-cell communications – cell signaling, Hormones, Pheromones and cell behavior
Defense mechanisms:
In plants: Herbivory, secondary metabolites
In animals: Innate and Adaptive immune systems

[06 Hrs]

Unit VI : Engineering perspectives of biological sciences:
Biology and engineering crosstalk – At cell level: Hybridoma technology
At tissue level: Plant Tissue Culture, Animal Tissue Culture; Tissue Engineering: Principles, methods and applications Introduction to Biomimetics and Biomimicry, nanobiotechnology.

[06 Hrs]

Reference Books :

- Lodish H, Berk A, Zipursky SL, et al. (2000) Molecular Cell Biology. W. H. Freeman.
- Lehninger, A. L., Nelson, D. L., & Cox, M. M. (2000). Lehninger principles of biochemistry. New York: Worth Publishers.
- Rao CNR, et.al. Chemistry of Nanomaterials: Synthesis, Properties and Applications.

- Eggins BR. (1006) Biosensors: An Introduction. John Wiley & Sons Publishers.
- Palsson B.O. and Bhatia S.N. (2009) Tissue Engineering. Pearson.
- Yoseph Bar-Cohen (2005). Biomimetics- Biologically Inspired Technologies
- Joseph D. Bronzino, John Enderle, Susan M. Blanchard (1999) Introduction to Biomedical Engineering.
- Routledge Taylor and Francis group (2012). Introduction to Bio-medical Engineeering technologies

Table 1.1: For Teachers: Additional topics to be discussed with students in accordance with relevant biological topics (in branch-wise manner)

Disease/ Disorder	Physiology	Diagnosis	Therapeutics		Medical procedure
			Biomaterials	Instrumentation	
Cardiovascular disease	Heart – electrical stimulation and mechanical pumping	ECG, Angiography	Stentsfor angioplasty	Heart lung machines	Angioplasty, By-pass surgery
Bone/skull injuries	Biomechanics of musculo-skeletal system	Medical imaging technologies Arthroscopy	Prosthetics	Arthroscope Biomechanics Prosthetics	Joint replacement Total hip Replacement reh abilitation engg
Kidney disorders	Functioning of Kidney	Medical imaging technologies	Filtration membranes	Dialyser	Dialysis

(CT-20018) Rapid Prototyping Practice using Object Oriented Programming

Teaching Scheme	**Examination Scheme**
Lectures: 1 Hrs / Week	Term work, Course Project: 60 marks
Laboratory : 2 Hrs/ Week	External Evaluation: 40 marks

Course Outcomes

At the end of the course, the student will be able to:

1. Design classes for given real life data and operations .
2. Design hierarchy of classes using inheritance.
3. Demonstrate ability to use OO concepts in designing an application.
4. Develop an application prototype using Object Oriented Programming.

Course Contents

Introduction: Objects, Classes, Messages, Methods, Constructors

[2 Hrs]

Object Oriented Design: Encapsulation, inheritance, Polymorphism, access specification, Abstract classes, Abstract methods, Interfaces

[5 Hrs]

GUI and Event Driven Programming: Windows, GUI basics, Layout, Menus, Events, Event handlers

[2 Hrs]

Other Concepts: Interfaces, packages, exception handling.

[3 Hrs]

Case studies in Object Oriented Design: Examples from real life projects

[2 Hrs]

The instructor is free to give a choice from any programming language like Python, C++, Java, etc.

Suggested List of Assignments

This is only a suggestive list. Instructor is encouraged to update the list of assignments and projects. The purpose of the assignments should be to lead to a meaningful project. The students should be encouraged to build different projects.

1. Design and implement a class representing a rectangle and operations like area, perimeter, change dimensions, report dimensions, on it. Write code to test your classes.
2. Design and implement classes representing a polygon shape, and derived classes like rectangle, and square with the same operations as mentioned in assignment above. Write code to test your classes.

3. Use the GTK or QT or similar library to create a window, add a canvas to it and draw a shape with mouse clicks.
4. Use the GTK or QT or similar library to create a window, add buttons representing rectangle and triangle, add a canvas, and draw a rectangle or triangle using mouse clicks, reusing the code you wrote in the first assignment.

Illustrative Mini Project

The project should be built using a version control system like git.

Build a graphical application on the lines of 'xfig' using Object Oriented Programming principles.

The application should enable one to draw various drawings, and save them in various formats like JPG, PDF, BMP, etc.

The drawing should have tools like rectangle, square, circle, ellipse, triangle, polygons of any shape and supported operations of coloring the borders and areas, erasing elements, changing size and shape, moving elements around, etc.

References

- "Learning Python", Mark Lutz, 5th Edition, 2013, O'Reilly Media; ISBN-10: 1449355730; ISBN-13: 978-1449355739
- "Beginning Java Programming: The Object-Oriented Approach", Bart Baesens, Aimee Backiel, Seppe vanden Broucke , 1st Edition, 2015, Wrox Publisher,
- "Object Oriented Programming Using C++ and Java" , E Balagurusamy, 1st Edition, 2017, McGraw Hill Education; ISBN-10: 1259006492 , ISBN-13: 978-1259006494
- A Sequence of Assignments to Teach Object-Oriented Programming: a Constructivism Design-First Approach
  https: //www.mii.lt/informatics_in_education/pdf/INFE016.pdf

(ICE(IF)-20002) Sensors and Automation

(Interdisciplinary Foundation Course–II)

Teaching Scheme	**Examination Scheme**
Lectures: 1 Hr / Week	Assignment/Quizzes : 40 marks
Practicals: 2 Hrs / Week	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Interpret the characteristics of the transducers/sensors
2. Select transducers/sensors for specific applications
3. Understanding of working principle of Programmable Logic Controller (PLC) and Distributed Control Systems (DCS)
4. Understanding the concept of Industrial Automation

Course Contents

Unit I: Basics of Sensors: Concepts and terminology of transducer, sensor, sensor classifications and characteristics (Static and dynamic), Working principle, characterization and applications of: strain gauges, LVDT, capacitive, RTD, thermocouple, thermistor, Solid-State, pressure, optical, chemical sensors, integration of sensors for IOT and Industry 4.0 applications.

[7 Hrs]

Unit II: Industrial Automation Industrial Automation: concept, automation components, necessity and working principle, block schematic of Programmable Logic Controller (PLC). Input & Output modules (AI, DI, AO, DO), Introduction to Ladder Programming, introduction to Distributed Control Systems (DCS). Industrial automation leads to Industrial IOT and Industry 4.0.

[7 Hrs]

List of Practicals

1. Case study /Characterization of RTD/semiconductor Temp IC
2. Characterization of level sensors
3. Characterization of strain gauge/ Displacement measurement using LVDT/ Encoders
4. Characterization of PH, Conductivity, color sensor
5. Introduction to PLC programming languages (ladder programming)
6. Ladder Programming for relay, coil, On/OFF, Sequencing of motors,
7. Ladder Programming with Timers/Counters
8. Ladder Programming for Pick and Place type of robotics application

Text Books

- B. C. Nakra and K. K. Choudhari, "Instrumentation Measurements and Analysis" by, Tata McGraw Hill Education, Second ed., 2004.
- C.D. Johnson, " Process Control Instrumentation Technology" by, Pearson Education Limited , eighth ed., 2014Reference Books

(CT- 20012) Theory of Computation

Teaching Scheme	**Examination Scheme:**
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
Tutorials: 1 hr / week	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Design Finite Automata/ Push Down Automata/Turing Machines as various representations for different language classes.
2. Apply transformation between multiple representations of automata/machines.
3. Distinguish different formal computing languages and classify their respective types.
4. Prove and disprove theorems establishing closure properties of formal languages.

5. Prove the decidablity and undecidability problems

Course Contents

Introduction: Automata, Computability, and Complexity, Strings and languages: symbol, alphabet, string/ word. Language - Definition, language states, difference between natural and formal language.

[4 Hrs]

Finite Automata: Formal definition of a finite automaton, Examples of finite automata, Formal definition of computation, Designing finite automata, the regular operations. Non-determinism: Formal definition of a nondeterministic finite automaton, Equivalence of NFAs and DFAs, Closure under the regular operations.

[10 Hrs]

Regular Expressions and Pumping Lemma: Regular Expressions: Formal definition of a regular expression, Equivalence with finite automata. Non regular Languages: The pumping lemma for regular languages.

[6 Hrs]

Context-Free Languages: Context-free Grammars: Formal definition of a context-free grammar, Examples of context-free grammars, Designing context-free grammars, Parse Trees, Ambiguity, Chomsky Normal Form. Pushdown Automata: Formal definition of a pushdown automaton,
Examples of pushdown automata, Equivalence with context-free grammars. Non-context-free Languages: The pumping lemma for context-free languages.

[8 Hrs]

The Church-Turing Thesis: Turing Machines: Formal definition of a Turing machine, Examples of Turing machines. Variants of Turing Machines: Multi-tape Turing machines, Nondeterministic Turing machines, Enumerators, Equivalence with other models. The Definition of Algorithm: Hilbert's problems, Terminology for describing Turing machines.

[8 Hrs]

Decidability: Decidable Languages: Decidable problems concerning regular languages, Decidable problems concerning context-free languages, The Halting Problem: The diagonalization method, the halting problem is undecidable, A Turing-unrecognizable language.

[4 Hrs]

Text Books

- “Introduction to the Theory of Computation”, Michael Sipser, 3rd Edition, 2013, Cengage Learning Publications, ISBN-13: 978-1133187790.
- “Introduction to Automata theory, Languages, and Computations”, John E Hopcroft, Rajeev Motwani, J D Ullman, 3rd Edition, 2009, Pearson Education Publisher, ISBN-10: 0321455363 / ISBN-13: 978-0321455369

Reference Books

- “Theory of computer science”, E. V. Krishnamurthy, 2004, Affiliated East Press Publications, ISBN-10: 038791255X / ISBN-13: 978-0387912554.
- "Automata and Computability", Dexter C. Kozen, 1997, Springer Verlag Publications, ISBN 0-387-94907-0.
- “Elements of the Theory of Computation”, Harry Lewis, Christos H. Papadimitriou, 2nd Edition, 1997, Prentice-Hall Publications, ISSN 0891-4516.
- “Introduction to Languages and Theory of Computations”, John Martin, 4th edition, 2010, McGraw-Hill Publications, ISBN 978–0–07–319146–1 / MHID 0–07–319146–9.

(CT-20013) Microprocessor Techniques

Teaching Scheme	**Examination Scheme:**
Lectures: 3 Hrs / Week	Assignment/Quizzes -40 marks
	End Semester Exam -60 marks

Course Outcomes

Students will be able to

1. Design memory organization
2. Explain 8086 architecture and its instruction set
3. Develop assembly language programs for the X86 microprocessor
4. Interface peripheral chips with respect to PPI, timer, serial communication and DMA controller
5. Describe the interrupt response of 8086 family processors

Course Contents

Memory Organization: Review of tri-state logic, buffers, latch, decoder, memory, and memory organization using typical RAM Chips. Evolution of microprocessor, Introduction to x86 microprocessor architecture, clock drivers, Memory interfacing, Memory Map, Address decoding logic.

[6 Hrs]

Instruction Set: x86 instruction encoding format, addressing modes and Instruction set, Assembly language programming, Assembler directives, Stacks and subroutines. General purpose register in IA-32 CPU and x86-64 CPU. Bus cycle, wait state, programming with string instructions, loop, rep.

[8 Hrs]

I/O Programming: I/O programming, Memory mapped I/O, I/O mapped I/O, Polled I/O, PPI 8255, Various operating modes of 8255, interfacing, and programming, 4x4 key matrix interfacing, Seven Segment display interfacing.

[6 Hrs]

Interrupts: 8086 Interrupt structure, ISR, PIC 8259 interfacing and programming.

[6 Hrs]

Timer and DMA: 8253 Timer, interfacing and programming. HOLD state and DMA, DMAC 8237.

[6 Hrs]

Serial Communication and Maximum Mode: Serial I/O, Asynchronous and Synchronous serial I/O, 8251 USART programming and interfacing, RS232C interface. Introduction to Maximum mode of 8086.

[8 Hrs]

Text Books

- "Microprocessors and its Interfacing (SIE)", Douglas Hall, S SS P Rao, 3rd Edition, 2017, McGraw Hill, ISBN-10: 9781259006159
- "Advanced Microprocessor and Peripherals", K M. Bhurchandi, A. K. Ray, 3rd Edition, 2017, McGraw Hill, ISBN-10: 1259006131

References

- "Microcomputer Systems: The 8086/88 Family", Liu, Gibson, 2nd Edition, 2005, PHI, ISBN: 978-81-203-0409-3
- "The 8086/88 Family: Design, Programming &Interfacing", John Uffenbeck, PHI, ISBN: 978-81-203-0933-3
- "Advanced MSDOS Programming", Ray Dunkon, 2nd Edition, BPB Publication, ISBN 1-55615-157-8
- "Assembly language for IBM PC", Kip Irvine, 2nd Edition, 1993, PHI,
- "Assembly language programming", Peter Abel, Pearson Education, 5th Edition, 2002, ISBN-10: 0137566107
- "Microprocessor 8086 : architecture, programming and interfacing", Sunil Mathur, PHI, 2011, ISBN : 9788120340879
- x86-64 Assembly Language Programming with Ubuntu. Version 1.1.40, Ed Jorgensen, January 2020

(CT-20016) Microprocessor Techniques Laboratory

Teaching Scheme	**Examination Scheme:**
Laboratory: 2 Hrs / Week	Term work : 50 Marks
	Practical Exam/Oral : 50 Marks

Laboratory Outcomes:

Students will be able to:

1. Develop assembly programs for the X86 microprocessor
2. Implement techniques for interfacing peripheral devices with microprocessor.

Suggested List of Experiments/Assignments

1. Implement Block data Transfer , Exchange Overlapping and non-overlapping blocks
2. Finding the smallest and largest number from a given list

3. Arranging a given list in ascending and descending order
4. 8/16/32 bit addition, 8/16/32 bit subtraction, multiplication, division
5. Finding the factorial of a given number
6. Finding the Fibonacci series of n terms
7. Performing - (a + b) * (c + d) and (a * b) + (c * d)
8. Finding the 1's and 2's complement of a given number
9. Check whether a number is even or odd
10. Multiplication by rotation and addition
11. Matrix addition and subtraction
12. Perform the following operations on a string: Find the length of a string and Reverse a string, Concatenate two strings and Compare two strings, Search a string for a given substring, Change the case of the string, Copy from one string to another string , Check whether the given string is a palindrome
13. Interface the following with the 8086: -
14. 8255, 8253, 8259, 8251
15. DAC - Generate Ramp, Triangle, Square, Sine waveform

Text Books

- "Microprocessors and its Interfacing (SIE)", Douglas Hall, S SS P Rao, 3rd Edition, 2017, McGraw Hill, ISBN-10: 9781259006159
- "Advanced Microprocessor and Peripherals", K M. Bhurchandi, A. K. Ray, 3rd Edition, 2017, McGraw Hill, ISBN-10: 1259006131

References

- "Microcomputer Systems: The 8086/88 Family", Liu, Gibson, 2ndEdition, 2005, PHI, ISBN: 978-81-203-0409-3
- "The 8086/88 Family: Design, Programming &Interfacing", John Uffenbeck, PHI, ISBN: 978-81-203-0933-3
- "Advanced MSDOS Programming", Ray Dunkon, 2nd Edition, BPB Publication, ISBN 1-55615-157-8
- "Assembly language for IBM PC", Kip Irvine, 2nd Edition, 1993, PHI,
- "Assembly language programming", Peter Abel, Pearson Education, 5th Edition, 2002, ISBN-10: 0137566107
- "Microprocessor 8086 : architecture, programming and interfacing", Sunil Mathur, PHI, 2011, ISBN : 9788120340879
- x86-64 Assembly Language Programming with Ubuntu. Version 1.1.40, Ed Jorgensen, January 2020

(CT-20014) Data Structures and Algorithms - II

Teaching Scheme	**Examination Scheme**
Lectures: 2 Hrs / Week	Assignments / Quizzes: 40 marks
	End Semester Exam:60 marks

Course Outcomes

Students will be able to:

1. Discuss, compare and analyze algorithms for various operations on different implementations of trees, graphs, matrices, heaps, etc.
2. Discuss, illustrate, and analyze properties of various data structures.
3. Design and implement data structures like trees, graphs, etc.
4. Implement an application of at least 1000 lines of code, demonstrating suitable choice of linear and non-linear data structures and algorithms.

Course Contents

Linear Data Structures: Recall - Static and Dynamic memory allocation, Arrays, Linked List, Stack, Queue. Time Complexity Analysis. Verification of programs, invariants, assertions, proof of termination.

[3 Hrs]

Trees: Basic terminology. Binary Tree: Properties of a Binary Tree, ADT Binary trees and its representations, concept of Non- Linear Data Structures, Difference between Linear and Non-Linear data structure**,** Binary tree traversals (recursive and non-recursive) and various operations. Binary Search Tree(BST): Properties, Insertion and deletion of nodes. Complexity Analysis of all operations.

[6 Hrs]

Priority queues and Heap: Priority Queues. Max and Min Heap. Operations on heap, Heapsort. Applications of trees. AVL Trees: Introduction, Properties, Balance Factor, Operations like insert, rotate and delete, Red Black Trees: Properties, Operations like insert, delete and rotate.

[5 Hrs]

Graph: Representation of graphs using adjacency matrix, adjacency list. Implementation of algorithms for traversals; implementing Kruskal's or Prim's or Single source shortest paths using Dijkstra's algorithm. Applications of graphs for problems like shortest path on a map.

[6 Hrs]

Sparse Matrices. Matrix operations like addition and multiplication. Sparse matrices concept. Different implementations of sparse matrices. Operations like addition and multiplication. Time complexity comparisons.

[4 Hrs]

Applications. Design of data structures, modules, functions and algorithms for applications like Huffman Coding, Implementing a ‘heap’ manager (malloc/free library), programs like ‘tree’, ‘tar’, ‘diff’, ‘grep’, ‘find’, ‘zip’, ‘unzip’, ‘nano’, ‘vim’, large matrix operations library, map applications, air traffic simulator etc

[4Hrs]

Text Books

- "Fundamentals of Data Structures in C", E. Horowitz, S. Sahni, S. Anderson-freed, Second Edition, 2008, University Press, ISBN 978-81-7371-605-8
- "The C Programming Language", B. Kernighan, D. Ritchie, Second Edition, 2015, Pearson Education India; ISBN 81-203-0596-5

Reference Books:

- "Data Structures using C", Y. Langsam, M. Augenstin and A. Tannenbaum, First Edition, 2002, Pearson Education Asia, ISBN 978-81-317-0229-1
- "C++: The Complete Reference", Herbert Schildt, Fourth Edition, 2002, The McGraw-Hill company, ISBN 0-07-222680-3
- "Fundamentals of Data Structures in C++", Ellis Horowitz, S. Sahni, D. Mehta, 2nd Edition, 2008, University Press, ISBN-10: 8173716064
- "An introduction to data structures with Applications", Jean-Paul Tremblay, Paul. G. Soresan, 2nd Edition, 1984, Tata Mc-Graw Hill International Editions, ISBN-0-07-462471-7

(CT-20017) Data Structures and Algorithms – II Laboratory

Teaching Scheme	**Examination Scheme**
Laboratory: 2Hrs / Week	Continuous evaluation: 50 Marks
	Mini Project: 25 marks
	End Semester Exam: 25 Marks

Course Outcomes

Course Outcomes

Students will be able to:

1. Discuss, compare and analyze algorithms for various operations on different implementations of trees, graphs, matrices, heaps, etc.
2. Discuss, illustrate, and analyze properties of various data structures.
3. Design and implement data structures like trees, graphs, etc.
4. Implement an application of at least 1000 lines of code, demonstrating suitable choice of linear and non-linear data structures and algorithms.

Note: Students will be expected to *reuse* the ADT list, stack, queue codes they have written in 'Data Structures and Algorithms – I Laboratory' course in III semester, whenever required.

Suggested List of Assignments:

1. Write the following functions for a binary search tree implementation: Searches the maximum value in the tree, preorder traversal without using recursion, Search the string in the tree and returns a pointer to the node, print the binary tree so that it looks like a tree.
2. Write code to list leaf nodes, non-leaf nodes and level of all nodes in a given binary tree.
3. Write a code for level order traversal of a binary tree with and without stack.

4. Start with an empty AVL tree and perform series of insertions like : December, January, April, March, July, August, October, February, November, May, June. Display the tree.
5. Implement a sparse matrix with operations like initialize empty sparse matrix, insert an element, sort a sparse matrix on row-column, add two matrices and return the result as a matrix, transpose a matrix, etc.
6. Develop C functions to insert and delete into/from a max heap under the assumption that a dynamically allocated array is used, the initial capacity of this array is 1, and array doubling is done whenever we are to insert into a max heap that is full.
7. Implement Heap sort algorithm on a set of records, with a specified key.
8. Write a graph implementation, using adjacency lists and demonstrate Djkstra's algorithm on it.
9. Write a program to read a map stored in a text file with (city1, city2, distance) comma separated values. Build a graph using this data. Print all pairs shortest paths information between all pairs of cities.
10. Implement DFS and BFS on a Graph.
11. Write a program to find all connected components of a Graph, on a map specified in a text file as *Source, Destination, distance* comma separated values.
12. Develop a hash table implementation in which overflows are resolved using chaining. Read a set of records from a file, insert them into hash table, then perform a set of searches using use supplied data and show the search results.
13. Implement a dictionary using a sparse matrix data structure.

Mini-project: Write an application demonstrating your skills in defining a problem, writing down the requirements carefully, designing a modular solution with clear separation of abstract data types and their use, design of proper function prototypes and division of work among functions. The application can be a unix command re-implemented (e.g. find, tar, zip, unzip, nano, vim, etc), large matrix operations library, map applications, huffman coding, air traffic simulator etc.), an efficient memory allocator, a simple game using libraries like n-curses or SDL, games like sudoku or chess, a railways scheduler, etc.

(CT-20015) Data Communication

Teaching Scheme	Examination Scheme:
Lectures: 3 Hrs / Week	Assignment/Quizzes : 40 marks
	End Semester Exam : 60 marks

Course Outcomes

Students will be able to:

1. Explain Data Communications System and its components.
2. Identify the different types of network topologies and protocols required for a given real life scenario
3. Compare OSI model against TCP/IP
4. Apply basic protocols of computer networks in network design and implementation
5. Solve problems related to sub-netting and routing.

Course Contents

Introduction: Data Communications, Networks, Network Types, Internet, Protocols and Standards, Network Models: OSI, TCP/IP, Analog and Digital data, Periodic Analog Signal, Digital Signal, Transmission Impairments, Data Rate Limits, Performance, Analog and Digital Transmission.

[8 Hrs]

Bandwidth Utilization and Transmission Media: Multiplexing, Spread Spectrum, Guided Media and Unguided media, Fiber Channel Technologies: Fiber to the home (FTTH), Data Over Cable Service Interface Specification (DOCSIS)

[6 Hrs]

Switching: Switching and TCP/IP layers, Circuit Switched Networks, Packet Switching, Structure of Switch.

[3 Hrs]

Error Detection and Correction: Types of Errors, Redundancy, Detection Vs Correction, FEC Vs Retransmission, Coding, Modular Arithmetic, Block Coding, Linear Block Codes, Cyclic Codes, Checksum, Hamming Code.

[6 Hrs]

Data Link Control: DLC Services, Flow Control and Error Control Protocols, Protocols: stop-and-wait, Go-Back-N, Selective-Repeat, Piggybacking, HDLC, PPP.

[6 Hrs]

Medium Access, Ethernet and LAN: Random Access: ALOHA, CSMA, CSMA/CD, CSMA/CA , Controlled Access, Channelization, Ethernet Protocol, Standard Ethernet, Fast and Gigabit Ethernet, Connecting devices, Backbone networks, VLAN.

[8 Hrs]

Textbooks:

- B. A. Forouzan, "Data Communications and Networking", 5th Edition, Tata McGraw-Hill, 2013, ISBN-10: 1-25-906475-1
- William Stallings, "Data and computer Communication", 7thEdition, Pearson Education, 2003, ISBN-13: 978-0131006812, ISBN-10: 0131006819.

Reference Books

- Larry L. Peterson and Bruce S. Davie, "Computer Networks a systems approach", 5th Edition, Morgan Kaufmann an imprint of Elsevier, 2014, ISBN: 978-93-80501-93-2
- Alberto Leon Garcia and IndraWidjaja, "Communication Networks, Fundamental Concepts andKey Architectures", 2nd Edition, Tata McGraw-Hill. 2004, ISBN-10: 007246352X
- A S Tanenbaum, "Computer Networks", 4th Edition, Pearson Education, ISBN 9788177581652
- S. Keshav, "Engineering Approach to Computer Networks", Pearson Education, 1997, ISBN-13: 9780201634426

Online Resources:

- NPTL Course on Data Communication by H. S. Jamadagni, CEDT, IISc, Bangalore. Available at https: //nptel.ac.in/courses/106/108/106108098/
- DOCSIS and CMTS Architectural Overview. Available at https: //www.cisco.com/c/en/us/td/docs/ios/cable/configuration/guide/u10k_cmts_dsis_ov.html
- FTTH architecture white paper series. Available at https: //www.commscope.com/globalassets/digizuite/2597-ftth-architectures-wp-110964-en.pdf?r=1