<Margin 1 inch top, bottom, left and right>

College of Engineering, Pune

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

Department of Computer Engineering and Information Technology

Curriculum Structure & Detailed Syllabus (UG Program)

Second Year B.Tech. Computer Engineering

(Revision: A.Y. 2015-16, Effective from: A.Y. 2016-17)

INDEX

<Margin 1 inch top, bottom, left and right>

Program Education 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 provide opportunity to learn the latest trends in computer engineering and prepare for lifelong learning process.

III. To make the students aware of professional ethics of the software Industry and prepare them with basic soft skills essential for working in community and professional teams.

.

Program 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.

<Margin 1 inch top, bottom, left and right>

List of Abbreviations

Abbreviation	Title
S.P. P.U.	Savitribai Phule Pune University
A.Y.	Academic Year
BSC	Basic Science Course
EFC	Engineering Foundation Course
MLC	Mandatory Learning Course
ILOE	Institute Level Open Elective Course
SLC	Self Learning Course
HSMC	Humanities/Social Sciences/Management Course
LLC	Liberal Learning Course
SBC	Skill Based Course
PCC	Program Core Course
DEC	Department Elective Course
LC	Laboratory Course

<Margin 1 inch top, bottom, left and right>

Semester III [Odd Term]

Sr. No.	Course Type	Course Name	Teaching Scheme			Credits
			L	T	P	
1	BSC	Ordinary Differential Equations and Multivariate Calculus	2	1	-	**3**
2	BSC	Science of Living Systems	3	0	0	**3**
3	MLC	Professional Ethics & Values	1	-	-	**0**
4	HSMC	Innovation	1	-	-	**1**
5	SBC	Programming Laboratory	0	0	2	**1**
6	PCC1	Data Structures and Algorithms	**3**	1	0	**4**
7	PCC2	Digital Logic Design	3	0	0	**3**
8	PCC3	Discrete Structures and Graph Theory	3	0	0	**3**
9	SBC	Data Structures and Algorithms Laboratory	0	0	4	**2**
10	LC2	Digital Logic Design Laboratory	0	0	2	**1**
			16	**2**	**8**	**21**
		Total Academic Engagement and Credits	**26**			**21**

Semester IV [Even Term]

Sr. No.	Course Type	Course Name	Teaching Scheme			Credits
			L	T	P	
1	BSC	Vector Calculus and Partial Differential Equations	2	1	-	**3**
2	ILOE	Information Systems [For Other Departments]	3	-	-	**3**
4	PCC1	Theory of Computation	3	-	-	**3**
5	PCC2	Microprocessor Techniques	3	-	-	**3**
6	PCC3	Principles of Programming Languages	3	-	-	**3**
7	PCC4	Data Communication	3	-	-	**3**
8	LC1	Microprocessor Techniques Laboratory	-	1	2	**2**
9	SBC	Principles of Programming Languages Laboratory	-	-	2	**1**
			17	**2**	**4**	**21**
		Total Academic Engagement and Credits	**23**			**21**

<Margin 1 inch top, bottom, left and right>

Semester III (For Direct Second Year Admitted Diploma Students)

Sr. No.	Course Type	Course Name	Teaching Scheme			Credits
			L	T	P	
1	BSC	Linear Algebra and Univariate Calculus	4	1	0	5
2	BSC	Science of Living Systems	3	0	0	3
3	MLC	Professional Ethics & Values	1	0	0	0
4	HSMC	Innovation	1	0	0	1
5	SBC	Programming Laboratory	0	0	2	**1**
6	PCC1	Data Structures and Algorithms	**3**	1	0	**4**
7	PCC2	Digital Logic Design	3	0	0	**3**
8	PCC3	Discrete Structures and Graph Theory	3	0	0	**3**
9	SBC	Data Structures and Algorithms Laboratory	0	0	4	**2**
10	LC2	Digital Logic Design Laboratory	0	0	2	**1**
			18	**2**	**8**	**23**
		Total Academic Engagement and Credits	**28**			**23**

Semester IV (For Direct Second Year Admitted Diploma Students)

Sr. No.	Course Type	Course Name	Teaching Scheme			Credits
			L	T	P	
1	BSC	Multivariate Calculus and Differential Equations	4	1	-	5
2	ILOE	Information Systems [For Other Departments]	3	-	-	3
4	PCC1	Theory of Computation	3	-	-	**3**
5	PCC2	Microprocessor Techniques	3	-	-	**3**
6	PCC3	Principles of Programming Languages	3	-	-	**3**
7	PCC4	Data Communication	3	-	-	**3**
8	LC1	Microprocessor Techniques Laboratory	-	1	2	**2**
9	SBC	Principles of Programming Languages Laboratory	-	-	2	**1**
			19	**2**	**4**	**23**
		Total Academic Engagement and Credits	**27**			**23**

<Margin 1 inch top, bottom, left and right>

Semester-III

(MA) Ordinary Differential Equations and Multivariate Calculus

Teaching Scheme:
Lectures : 2 Hrs/week
Tutorial: 1 Hr/week

Examination Scheme:
T1 and T2: 20 Marks each
End-Sem Exam: 60 Marks

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)

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. **[10 Hrs]**

Unit II : Functions of several variables, level curves and level surfaces, partial and directional derivatives, differentiability, chain rule, local extreme values and saddle points, constrained optimization**.** **[05 Hrs]**

Unit III : 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. **[11 Hrs]**

Text Books:

- Maurice D. Weir, Joel Hass, Frank R. Giordano, "Thomas' Calculus", Pearson Education, 12^{th} Edition.
- Erwin Kreyszig, "Advanced Engineering Mathematics", Wiley eastern Ltd., 10^{th} Edition

Reference Books:

- Author name, "Title of the book in double quotes", Publisher, Edition, Year
- K.D Joshi, "Calculus for Scientists and Engineers", CRC Press.

<Margin 1 inch top, bottom, left and right>

- Sudhir Ghorpade and Balmohan Limaye, "A Course in Multivariate Calculus and Analysis", Springer Science and Business Media.
- George Simmons, "Differential Equations with Applications and Historical notes", Tata Mc-Graw Hill publishing company Ltd, New Delhi.
- C.R. Wylie, " Advanced Engineering Mathematics" , McGraw Hill Publications, New Delhi
- Peter V. O' Neil, "Advanced Engineering Mathematics", Thomson Brooks / Cole, Singapore, 7th edition

(CT) Programming Laboratory

Teaching Scheme:
Laboratory : 2 hours per week

Examination Scheme:
Practical/Oral Exam: 50 marks
Term work: 50 marks

Course Outcomes:

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

1. Understand, analyze and create web page using HTML and file handling
2. Understand basic programming language concepts, particularly Java and object-oriented concepts
3. Implementation of various operations on Stack, Queue and List.
4. Implement various searching and sorting algorithms.
5. Acquire practical knowledge of android technology for project development

Suggested List of Assignments:

1. Design a student registration form using HTML and write the data to a file using CGI scripts
2. Implement stack and list data types using C++/Java.
3. Demonstrate the use of List, and Queue type using Standard template library of C++/Java
4. Implementing stack and list data types in C++/Java using Eclipse IDE.
5. Implement binary search and selection sort algorithm in C++/Java using Eclipse IDE.
6. Design a text based normal and scientific calculator using C++/Java in Eclipse with help of built in libraries for advanced math operation
7. Design any game in android using Eclipse or android studio. Student can use any library as per requirement. (Mini Project)

(CT) Data Structures And Algorithms

Teaching Scheme:
Lectures : 3 Hrs/week
Tutorial: 1 Hr/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

<Margin 1 inch top, bottom, left and right>

Course Outcomes:

Students will be able to:

1. Write neat code 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 clearly different possible solutions to a program and select the most efficient one.
4. Write an application requiring an effort of at least 1000 lines of code to demonstrate a good working solution.
5. Demonstrate the ability to write reusable code and abstract data types in C, using object-based way of thinking.

Unit I : Introduction: Data. Data types. Object, data structure and abstract data types (ADT). Characteristics of an algorithm. Analyzing programs. Frequency count. Time and space complexity. Big 'O' and ΄Ω΄΄ notation. Best. average and worst cases. Dangling pointers and garbage memory. **[4 Hrs]**

Unit II : Arrays. Files and Searching: Searching: linear and binary search algorithm. Hashing: hashing functions, chaining, overflow handling with and without chaining, open addressing: linear. quadratic probing. Files handling: text and binary files, use of various libraries for handling files. **[6 Hrs]**

Unit III : 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. Recursion and stacks. Problems like maze and knight's tour. **[6 Hrs]**

Unit IV : Lists: List as ADT. Concept of linked organization of data against linked list. Singly linked list, doubly linked list, circular linked list. Representation & manipulations of polynomials/sets using linked lists. Dynamic memory management. Representation of sparse matrix. Addition and transpose of sparse matrix. **[8 Hrs]**

Unit V : Trees and Graphs: Basic terminology. Binary trees and its representation. Binary tree traversals (recursive and non recursive) and various operations. Insertion and deletion of nodes in binary search tree. Representation of graphs using adjacency matrix, adjacency list. Implementation of algorithms for traversals; implementing Kruskal's. Prim's algorithms. Single source shortest paths using Djkstra's algorithm. Applications of graphs and trees. **[8 Hrs]**

Unit VI : Time Complexity Analysis, Algorithm Design: Verification of programs, invariants, assertions, proof of termination. Best, Average and Worst case analysis of: binary search, quick sort, merge sort, insertion sort, hashing techniques, sparse matrix algorithms. Designing data structures for specific applications. **[8 Hrs]**

<Margin 1 inch top, bottom, left and right>

Text Books:

- E. Horowitz, S. Sahni, S.Anderson-freed, "Fundamentals of Data Structures in C", Second Edition, University Press, ISBN 978-81-7371-605-8
- B. Kernighan, D. Ritchie, "The C Programming Language", Prentice Hall of India, Second Edition, ISBN 81-203-0596-5
- Y. Langsam, M. Augenstin and A. Tannenbaum, "Data Structures using C", Pearson Education Asia, First Edition, 2002, ISBN 978-81-317-0229-1

Reference Books:

- Ellis Horowitz, S. Sahni, D. Mehta "Fundamentals of Data Structures in C++", Galgotia Book Source, New Delhi 1995 ISBN 16782928
- Jean-Paul Tremblay, Paul. G. Soresan, "An introduction to data structures with Applications", Tata Mc-Graw Hill International Editions, 2nd edition 1984, ISBN-0-07-462471-7

(CT) Digital Logic Design

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem 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. Learn and understand the basic concepts of combinational logic devices and apply the concepts in designing them.
3. Learn and understand the fundamentals of sequential logic devices and apply the concepts in designing them.
4. Apply and design the logical devices by using all these concepts along with implementation knowledge of hardware and peripheral design.

Unit I : Introduction to Number systems and codes : Binary number systems , Signed binary numbers, Binary arithmetic,1's and 2's complement, Octal number system, hexadecimal number system, Introduction to gates, 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]**

Unit II : Design of Combinational Logic Circuits: Modular combinational logic elements, Overview & implementation of multiplexer/ demultiplexer, Implementation of Combinational

<Margin 1 inch top, bottom, left and right>

Logic Circuits using mux / demux, Decoders, Encoders, Priority encoders. Design of Integer Arithmetic Circuits using Combinational Logic: Integer adders, 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 **[8 Hrs]**

Unit III : Design of 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]**

Unit IV: 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]**

Unit V : Algorithm State Machines: ASM charts, notation ,RTL notation and implementation design of simple controller, multiplexer controller method. **VHDL**: Introduction to HDL, VHDL-Library. **[5 Hrs]**

Unit VI Memories: Random access memory, TTL RAM cell, parameter read write cycles, ROMs EPROM,MOS-static RAM cell, dynamic RAM cell, refreshing, memory cycles. **[5 Hrs]**

Text Books:

- M Morris Mano "Digital Design" 3rd Edition Prentice Hall 2001 ISBN-10 / ASIN: 0130621218 ISBN-13 / EAN: 9780130621214
- R.P. Jain, "Modern Digital Electronics", 3rd Edition, Tata McGraw-Hill, 2003, ISBN 0 – 07 – 049492 – 4
- A.P. Malvino, D. P. Leach and G.Saha, "Digital Principles and Applications," 7/e, McGraw Hill, 2010.

Reference Books:

- Wakerly Pearon, "Digital Design: Principles and Practices", 3rd edition, 4th reprint, Pearson Education, 2004.
- A. Anand Kumar, "Fundamentals of digital circuits" 1st edition, PHI publication, 2001.
- Mark Bach, "Complete Digital Design", Tata MCGraw Hill, 2005.
- Stephen Brown, "Fundamentals of digital logic design with VHDL" 1st edition, TMH Publication 2002.

(CT) Discrete Structures and Graph Theory

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

<Margin 1 inch top, bottom, left and right>

Course Outcomes:

Students will be able to:

1. Explain basic terminology, formal logic, proofs, sets, relations, functions, recursion
2. Use formal logic proof and logical reasoning to solve problems
3. Relate the ideas of mathematical induction to recursion and recursively defined structures
4. Solve problems based on graphs, trees and related algorithms
5. Relate, interpret and apply the concepts to various areas of computer science

Unit I : Set Theory , Logic and Proofs : Propositions, Conditional Propositions, Logical Connectivity, Propositional calculus, Universal and Existential Quantifiers, First order logic, Proofs: Proof Techniques, Mathematical Induction. Set, Combination of sets, Finite and Infinite sets, Un-countably infinite sets, Principle of inclusion and exclusion , strong Induction **[6 Hrs]**

Unit II : 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 **[8 Hrs]**

Unit III : Number Theory: Basics of Modulo Arithmetic, Basic Prime Number Theory, GCD, LCM, Divisibility, Euclid's algorithm, Factorization, Chinese Remainder Theorem Fields: Naturals, Integers, Rationals, Reals, Complex Numbers Properties of operations: associative, commutative, distributive, identity, inverse **[6 Hrs]**

Unit IV : 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, Generating Permutations and Combinations **[6 Hrs]**

Unit V : 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, graph coloring. Trees, rooted trees, path length in rooted trees, binary search trees, spanning trees and cut set, theorems on spanning trees, cut sets , circuits, minimal spanning trees, Kruskal's and Prim's algorithms for minimal spanning tree. **[8 Hrs]**

Unit VI : 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:

<Margin 1 inch top, bottom, left and right>

- C. L. LIU, “Elements of Discrete Mathematics”, 2nd Edition, Tata McGraw-Hill, 2002, ISBN: 0-07-043476-X.
- G. Shanker Rao, ”Discrete Mathematical Structures”, New Age International, 2002, ISBN: 81-224-1424-9

Reference Books:

- Lipschutz, Lipson, Discrete Mathematics, 2nd Edition, Tata McGraw-Hill, 1999, ISBN 0-07- 463710--X.
- V. K. Balakrishnan, Graph Theory, TMH (Recommended for Graph), ISBN 0-07-058718-3
- B. Kolman, R. Busby and S. Ross, “Discrete Mathematical Structures”, 4th Edition, Pearson Education, 2002, ISBN 81-7808-556-9
- J. Tremblay, R. Manohar, “Discrete Mathematical Structures with application to Computer Science”, McGraw-Hill, 2002 ISBN 0-07-065142-6 (Recommended for prepositional Calculus)
- Kenneth H. Rosen: Discrete Mathematics and Its Applications, 5th Edition, Tata McGraw-Hill, 2003, ISBN 0-07-053047-5

(CT) Data Structures And Algorithms Laboratory

Teaching Scheme:
Laboratory : 4 hours per week

Examination Scheme:
Continuous evaluation: 50 Marks
Mini Project: 20 marks
End Semester Exam: 30 Marks

Course Outcomes:

Shared with the theory course: “Data Structures and Algorithms”

List of Assignments:

1. Write any 1 program showing your indentation and formatting skills. The program need not have any meaning, but it should have following constructs in C: if, while, for, switch, One nesting of 3 levels, = , * , scanf, printf, variables declaration
2. Create an account on typeracer.com. Participate in some race and show a speed of at least 40 words per minute.
3. Draw a diagram of data structures created by given code using a tool like xfig.
4. 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
5. 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.

<Margin 1 inch top, bottom, left and right>

6. 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.
7. 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.
8. 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.
9. Write an data type called "Integer". The data type should represent integers of unlimited length.
10. 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.
11. Write the following functions for a binary search tree implementation: Searches the maximum value in the tree, preorder traversal without using recursion, Search the str in the tree and returns a pointer to the node, print the binary tree so that it lookks like a tree
12. Write a graph implementation, using adjacency lists.
13. Mini-project: Write an application of your own demonstration 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. cut, find, tar, fdupes, bc, etc.), reimplementation of C library functions, memory allocator, a simple game using libraries like n-curses or SDL, games like sudoku or chess, or an application to manage institutions like hospitals, colleges, shops, etc.

(CT) Digital Logic Design Laboratory

Teaching Scheme:
Laboratory : 2 hours per week

Examination Scheme:
Practical/Oral Exam: 50 marks
Term Work: 50 marks

List of Assignments:

1. Implementation of Boolean function using Gates

<Margin 1 inch top, bottom, left and right>

2. Code converters:
    - Binary to gray
    - Gray to binary
    - Excess – 3 code to BCD
    - BCD to Excess – 3 code.
3. Design of half adder, full adder.
4. Design of half subtract or , full subtract or.
5. K-map examples implementation
6. Quine-Mc'clusky examples implementation.
7. Design of :
    - 3 bit odd Parity Checker
    - 4 bit odd Parity Checker
    - 3 bit even Parity Checker
    - 4 bit even Parity Checker
8. Implementation of Multiplexer and Demultiplexer.
9. BCD adder using 4 bit adder IC.
10. Study of flip flops-
    - RS flip-flop
    - D flip-flop
    - T flip-flop
    - J-K flip-flop
11. Design of Synchronous Counter.
12. Design of Asynchronous counter.
13. Design of up / down counters.
14. Design of Sequence generator.
15. Design of Ring counter.
16. Design of Johnson Counter
17. Study Assignment on VHDL programming.

(CT) Vector Calculus and Partial Differential Equations

Teaching Scheme:
Lectures : 2+1 hrs/week
Credits: 2-1-0-3 (L-T-P-C)

Examination Scheme:
Internal Test 1 – 20 marks
Internal Test 2 – 20 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

<Margin 1 inch top, bottom, left and right>

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.)
6. organize and present thoughts. (To measure this outcome, questions may asked to write summaries and short notes on a given topic.)

Unit I : 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. **[9 Hrs]**

Unit II : Partial differential equations with separation of variables, boundary value problems: vibrations of a string, heat equation, potential equation, vibrations of circular membranes. **[10 Hrs]**

Unit III : Laplace Transforms, its properties , Unit step function, Dirac delta functions, Convolution Theorem, periodic functions, solving differential equations using Laplace transform. **[7 Hrs]**

Text Books:

- Thomas' Calculus (12^{th} edition) by Maurice D. Weir, Joel Hass, Frank R. Giordano, Pearson Education.
- Advanced Engineering Mathematics (10^{th} edition) by Erwin Kreyszig, Wiley eastern Ltd.

Reference Books:

- Lipschutz, Lipson, Discrete Mathematics, 2nd Edition, Tata McGraw-Hill, 1999, ISBN 0-07- 463710--X.
- V. K. Balakrishnan, Graph Theory, TMH (Recommended for Graph), ISBN 0-07-058718-3
- B. Kolman, R. Busby and S. Ross, "Discrete Mathematical Structures", 4th Edition, Pearson Education, 2002, ISBN 81-7808-556-9
- J. Tremblay, R. Manohar, "Discrete Mathematical Structures with application to Computer Science", McGraw-Hill, 2002 ISBN 0-07-065142-6 (Recommended for prepositional Calculus)

<Margin 1 inch top, bottom, left and right>

- Kenneth H. Rosen: Discrete Mathematics and Its Applications, 5th Edition, Tata McGraw-Hill, 2003, ISBN 0-07-053047-5

(CT) OPEN ELECTIVE - Information Systems

Teaching Scheme:
Lectures : 3 hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

After studying this course it will develop ability to:

1. Analyze functional and non-functional requirements to produce a system architecture that meets those requirements
2. Understand and apply process and methodology in building the application
3. Create design models using known design principles (e.g. layering) and from various view points (logical, physical etc.)
4. Explain and justify all the design choices and tradeoffs done during the application's development

Unit I : Introduction: Define and understand the term information systems (IS). Technology, people, and organizational components of an information system, various types of information systems, nature of information systems in the success and failure of modern organizations, Understand and plan for the future of managing IS. Information systems for automation, organizational learning and strategic support, Formulate and present the business case for a system. **[5 Hrs]**

Unit II : Database Management and Internet: Importance of databases in modern organizations, Working of database management systems, Database design, Query Processing, how organizations are getting the most from their investment in database technologies. Role of telecommunications in organizations, Types of computer networks, Extranets, Intranets, Working of Internet, Basic Internet services, World Wide Web. **[6 Hrs]**

Unit III : Information Systems Development and Acquisition: Process used by organizations to manage the development of information Systems. Major phases of the systems development life cycle: systems identification, selection, and planning; system requirement specifications; system design; system implementation; and system maintenance. Software prototyping, rapid application development, object-oriented analysis and design methods of systems development and their strengths and weaknesses, Factors in building a system in-house, along with situations, three system development options: external acquisition, outsourcing, and end-user development. **[8 Hrs]**

Unit IV : Organizational Information Systems: Characteristics of the operational, managerial, and executive levels of an organization, decision support systems, expert systems, office automation systems, collaboration technologies. **[7 Hrs]**

<Margin 1 inch top, bottom, left and right>

Unit V : Electronic Commerce: Business to Customer e-commerce, Business to Business e-commerce, Customer to Customer e-commerce, Advantages and disadvantages of e-commerce, E-Commerce System Architecture, Payment schemes in e-commerce, Cash transactions in e-commerce, e-commerce applications. **[6 Hrs]**

Unit VI : Information Systems Ethics, Computer Crime, and Security: Impact of computer ethics on information systems, Issues associated with information privacy, accuracy, property and accessibility, computer crime and list several types of computer crime, computer virus, worm, Trojan horse, and logic or time bomb, various methods for providing computer security, IT Act 2000. **[6 Hrs]**

Text Books:

- "Information Systems Today, Managing in the Digital World" , Third Edition by Leonard M. Jessup; Joseph S. Valacich, Publisher: Prentice Hall
- "Introduction to Information Technology", V. Rajaraman, PHI

Reference Books:

- "Information Systems Management in Practice" Barbara C. McNurlin, Ralph H. Sprague, and Publisher: Pearson Education.

(CT) Theory Of Computation

Teaching Scheme:
Lectures : 3 hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Identify different formal language classes and their relationships
2. Design grammars and recognizers for different formal languages
3. Construct finite state machines and the equivalent regular expressions
4. Prove the equivalence of languages described by finite state machines and regular expressions.

Unit I : Introduction: Automata, Computability, and Complexity, Strings and languages: symbol, alphabet, string/ word. Language - Definition, language states, difference between natural and formal language. **[6 Hrs]**

Unit II : Finite Automata: Formal definition of a finite automaton, Examples of finite automata, Formal definition of computation, Designing finite automata, The regular operations. Non-

<Margin 1 inch top, bottom, left and right>

determinism: Formal definition of a nondeterministic finite automaton, Equivalence of NFAs and DFAs, Closure under the regular operations. **[8 Hrs]**

Unit III : Regular Expressions and Pumping Lemma: Regular Expressions: Formal definition of a regular expression, Equivalence with finite automata. Nonregular Languages: The pumping lemma for regular languages. **[6 Hrs]**

Unit IV : Context-Free Languages: Context-free Grammars: Formal definition of a context-free grammar, Examples of context-free grammars, Designing context-free grammars, 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]**

Unit V : 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. **[6 Hrs]**

Unit VI : 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. **[6 Hrs]**

Text Books:

- Michael Sipser, "Introduction to the Theory of Computation", Cengage Learning Publications, 3rd Edition, 2013.
- John E Hopcroft, Rajeev Motwani, J D Ullman, "Introduction to Automata theory, Languages, and Computations", Pearson Education Publisher, 3rd Edition, 2009

Reference Books:

- E. V. Krishnamurthy, "Theory of computer science", Affiliated East Press Publications, 2004.
- Dexter C. Kozen, Automata and Computability, Springer Verlag Publications, 1997.
- Harry Lewis, Christos H. Papadimitriou, "Elements of the Theory of Computation," Prentice-Hall Publications, 2nd Edition, 1997.
- John Martin, "Introduction to Languages and Theory of Computations", McGraw-Hill Publications, 4th edition, 2010

(CT) Microprocessor Techniques

Teaching Scheme:
Lectures : 3 hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

<Margin 1 inch top, bottom, left and right>

Course Outcomes:

Students will be able to:

1. Students will be able to explain concepts of memory organization and design
2. Students will be able to explain x86 architecture
3. Students will have developed skills to develop an assembly language programs for the X86 microprocessor
4. Students will be able to interface peripheral chips with respect to timer, interrupts, serial communication and DMA controller.

Unit I : Review of tri-state logic, buffers, decoders, memory and memory organization using typical RAM Chips. Evolution of microprocessor, Introduction to x86 microprocessor architecture, clock drivers and buffers. Memory interfacing, Memory Map, Address decoding logic. **[6 Hrs]**

Unit II : x86 instruction encoding format, addressing modes and Instruction set, Assembly language programming, Assembler directives, Stacks and subroutines. Bus cycle, wait state, programming with string instructions, loop, rep, architecture-dependent implementations of various constructs and mechanisms of high level languages. **[8 Hrs]**

Unit III : I/0 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]**

Unit IV : 8086 Interrupt structure, ISR, PIC 8259 interfacing and programming, 8253 Timer. **[6 Hrs]**

Unit V : 8279 Keyboard Display Controller, interfacing and programming, HOLD state and DMA, DMAC 8237. **[6 Hrs]**

Unit VI : Serial I/0, Asynchronous and Synchronous serial I/0, 8251 USART programming and interfacing, RS232C interface, Introduction to Maximum mode of 8086. **[8 Hrs]**

Text Books:

- Douglas Hall, “Microprocessors and Interfacing”, 2nd edition, 1992, McGraw-Hill, ISBN-0-07-100462-9
- John Uffenbeck, “The 8086/88 Family: Design, Programming & Interfacing”, PHI, ISBN: 978-81-203-0933-3
- A.Ray, K.Bhurchandi,”Advanced Microprocessors and peripherals: Arch, Programming & Interfacing”,Tata McGraw Hill, 2004,ISBN 0-07-463841-6

Reference Books:

- Liu, Gibson, “Microcomputer Systems: The 8086/88 Family”, 2ndEdition, PHI, 2005, ISBN: 978-81-203-0409-3

<Margin 1 inch top, bottom, left and right>

- Ray Duncan, “Advanced MSDOS Programming”, 2ndEdition, BPB Publication,ISBN 1-55615-157-8
- Kip Irvine,”Assembly language for IBM PC”, PHI, 2ndEdition, 1993
- Peter Abel,”Assembly language programming”, Pearson Education, 5thEdition, 2002, ISBN-10: 0137566107

(CT) Principles of Programming Languages

Teaching Scheme:
Lectures : 3 hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

1. Draw the control flow of a program.
2. Understand the storage concepts in a simple program.
3. Program using basic concepts of OO languages i.e. objects, encapsulation, data hiding etc.
4. Program using advanced concepts of OO languages i.e. associations, packages, interfaces, exception handling etc.
5. Work with functional, Logic programming paradigms.

Unit I : Introduction: Role of programming languages, need to study programming languages, characteristics of good programming languages, Introduction to various programming paradigms: Procedural, object-oriented, logic and functional, concurrent programming.

Data Types: Properties of structured and non-structured data types and Objects, variables, constants, Derived and abstract data types, declaration, type checking. Binding and binding times, type conversion, scalar data type, composite data types, Implementation and Storage representation of data types and control flow statement. **[6 Hrs]**

Unit II : Procedures: Procedure call and return, recursive subprogram, Different parameter passing methods, Lifetime of variables, Scope rules: Static and Dynamic scope, Referencing environment: activation records (Local, Non local and Global), Storage management (static and Dynamic), Exceptions and exception handling. **[8 Hrs]**

Unit III : Object Oriented Programming: Design Principles: Objects, classes, Messages and methods, Implementation of Object-oriented Programming objective. **[6 Hrs]**

Unit IV : Object oriented programming with Java: Program structure, Object and class declarations, constructors, inheritance, polymorphism, access specification, interfaces, packages, exception handling, file I/O, GUI development, socket programming. **[8 Hrs]**

<Margin 1 inch top, bottom, left and right>

Unit V : Logic Programming: Logic programming language model, logical statements, resolution, unification, search structures: backward and forward, Applications of logic programming. Functional Programming: Introduction to functional programming, Lambda calculus: Ambiguity, free and bound identifiers, reductions, typed lambda calculus, application of functional programming. **[6 Hrs]**

Unit VI : Concurrent programming and functional programming: Basic concepts of Concurrent Programming: processes, synchronization primitives, safety and live-ness properties, Parallelism in Hardware, streams, concurrency as interleaving, safe access to shared data. Functional Programming. **[6 Hrs]**

Text Books:

- 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
- Herbert Schilt, “JAVA Complete Reference”, 7th Edition, Tata McGraw Hill, ISBN: 9780070636774
- Mark Lutz, “Learning Python”, 2nd Edition, O'reilly, ISBN: 978-0-596-00281-7
- Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, “C++ Primer”, 3rd Edition, Addison Wesley Professional, ISBN-10: 0201824701

Reference Books:

- Scbesta R., "Concepts Of Programming Languages", 4th Edition, Pearson Education, ISBN- 81-7808-161-X
- Ghezzi C, Milano P., Jazayeri M., "Programming Languages Concepts", 3rd Edition, John Wiley and Sons Pvt. Ltd (WSE), ISBN - 0195113063
- M. Ben Ari, “Principles of Concurrent Programming, 1989
- Eckel B., "Thinking in Java", 3rd Edition, Pearson Education,
- T. W. Pratt , "Programming Languages", 2nd Edition ,Prentice-Hall Of India, ISBN 81 - 297 - 0524 - 9
- Michael L. Scott “Programming Language Pragmatics”, ELSEVIER Publication, ISBN: 81-8147-370-1

(CT) DATA COMMUNICATION

Teaching Scheme:
Lectures : 3 Hrs/week

Examination Scheme:
Assignment/Quizzes – 40 marks
End Sem Exam - 60 marks

Course Outcomes:

Students will be able to:

<Margin 1 inch top, bottom, left and right>

1. After completing this course the student must demonstrate the knowledge and ability to: Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
5. Understand and building the skills of sub-netting and routing mechanisms.
6. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Unit I : Introduction: Data Communication, Networks, Internet, Protocols and Standards, Network Models: OSI, TCP/IP, Analog and Digital data, Periodic Analog Signal, Digital Signal, Transmission Impairments, Data Rate Limits, Performance. Signal Conversion: digital-to-digital, Analog-to-Digital, Analog-to-Analog, Digital-to-Analog Conversion. **[8 Hrs]**

Unit II : Bandwidth Utilization and Transmission Media: Multiplexing, Spread Spectrum, Guided Media and Unguided media. **[5 Hrs]**

Unit III : Switching: Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks, Structure of Switch. **[3 Hrs]**

Unit IV : 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]**

Unit V : Data Link Control: Framing, Flow Control and Error Control Protocols, Protocols: stop-and-wait, Go-Back-N, Selective-Repeat, Piggybacking , HDLC,PPP. **[6 Hrs]**

Unit VI : Medium Access, Ethernet and LAN: Random Access: ALOHA, CSMA, CSMA/CD, CSMA/CA, Controlled Access, Channelization, IEEE standards, different Ethernets, Connecting devices, Backbone networks, VLAN. **[8 Hrs]**

Text Books:

- B. A. Forouzan, “Data Communications and Networking”, 5th Edition, Tata McGraw-Hill, 2013, ISBN-10: 1-25-906475-1
- Alberto Leon Garcia and Indra Widjaja, “Communication Networks, Fundamental Concepts and Key Architectures”, 2nd Edition, Tata McGraw-Hill. 2004, ISBN-10: 007246352X

Reference Books:

- William Stallings, “Data and computer Communication”, 7th Edition, Pearson Education, ISBN-81-297-0206-1
- A S Tanenbaum, “Computer Networks”, 4th Edition, Pearson Education, ISBN 9788177581652

<Margin 1 inch top, bottom, left and right>

- S. Keshav , “Engineering Approach to Computer Networks”, Pearson Education, 1997, ISBN-13: 9780201634426
- J.F. Kurose and K. W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet” , 2nd Edition, Pearson, 2003, ISBN-13: 9780201976991

(CT) Microprocessor Techniques Laboratory

Teaching Scheme
Credits: 0-1-2-3 (L-T-P-C)
Laboratory : 2 hours per week,
Tutorial 1 Hour per week

Examination Scheme :
Term work - 50 Marks
Practical Exam/Oral – 50 Marks

Laboratory Outcomes:

At the end of the laboratory work, students will demonstrate the ability to:

1. Students will demonstrate skill of developing assembly programs for the X86 microprocessor
2. Students will demonstrate techniques for interfacing I/O devices to microprocessor.

List of Experiments/Assignments:

Assignments based on

X86 Assembly language Program development using

1. Data transfer, arithmetic and control instructions
2. Assembler directives
3. Stack and Subroutine
4. String instructions

Interfacing, programming of peripheral chips 8255/8259/8253/ 8251/8279/DAC.

Text Books:

- Carl Hamacher, Zvonko Vraesic and Safwat Zaky, Computer Organisation, 5thEdition, 2002, McGraw-Hill, ISBN 0-07-120411-3
- Douglas Hall, “Microprocessors and Interfacing”, 2nd edition, 1992, McGraw-Hill, ISBN-0-07-100462-9
- John Uffenbeck, “The 8086/88 Family: Design, Programming & Interfacing”, PHI, ISBN: 978-81-203-0933-3
- A.Ray, K.Bhurchandi,”Advanced Microprocessors and peripherals: Arch, Programming & Interfacing”,Tata McGraw Hill, 2004,ISBN 0-07-463841-6

Reference Books:

<Margin 1 inch top, bottom, left and right>

- William Stallings, "Computer Organization And Architecture Designing For Performance", Pearson Education, 8thEdition, 2010, ISBN: 978-81-317-3245-8
- Liu, Gibson, “Microcomputer Systems: The 8086/88 Family”, 2ndEdition, PHI, 2005, ISBN: 978-81-203-0409-3
- Ray Duncan, “Advanced MSDOS Programming”, 2ndEdition, BPB Publication,ISBN 1-55615-157-8
- Kip Irvine, “Assembly language for IBM PC”, PHI, 2ndEdition, 1993
- Peter Abel, “Assembly language programming”, Pearson Education, 5thEdition, 2002, ISBN-10: 0137566107

(CT) Principles of Programming Languages Laboratory

Teaching Scheme
Laboratory : 2 hours per week

Examination Scheme :
Continuous evaluation: 55 Marks
Mini Project: 15 marks
End Semester Exam: 30 Marks

Laboratory Outcomes:

Shared with the theory course: Principles of Programming Languages

List of Experiments/Assignments:

1. Assignment to understand creation of activation record.
2. Assignment to write program in OO language to understand concept of data abstraction and encapsulation.
3. Assignment to write program in OO language to understand concept of class inheritance and polymorphism.
4. Assignment to write program in OO language to understand concept of exception handling and file I/O.
5. Assignment to write an expert system using functional programming language.
6. Assignment to write a program to demonstrate use of logical programming language.
7. Assignment to write a program to demonstrate threads.
8. Assignment to write a program to demonstrate use of functional programming language.

(MA) Linear Algebra and Univariate Calculus

Teaching Scheme:
Lectures : 4 hrs/week
Tutorial: 1 hr/week

Examination Scheme:
Internal Test 1 – 20 marks
Internal Test 2 – 20 marks
End Sem Exam - 60 marks

<Margin 1 inch top, bottom, left and right>

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. Row and Column spaces, rank. Applications to systems of linear equations. **[14 Hrs]**

Unit II : Linear mappings, representation by matrices, 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. **[10 Hrs]**

Unit IV : Integrals as limits of Riemann sums, fundamental theorem of calculus, surface area, integrals by special techniques: reduction formulae, arc length, solids of revolution, improper integrals, tests for convergence, Gamma and Beta functions **[12 Hrs]**

Text Books:

- Thomas' Calculus (12^{th} edition) by Maurice D. Weir, Joel Hass, Frank R. Giordano, Pearson Education.
- Advanced Engineering Mathematics (10^{th} edition) by Erwin Kreyszig, Wiley eastern Ltd.

Reference Books:

- Introduction to Linear Algebra (2^{nd} edition) by Serge Lang, Springer.
- Elementary Linear Algebra (10^{th} edition) by Howard Anton and Chris Rorres, John Wiley and sons.

<Margin 1 inch top, bottom, left and right>

- 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

(MA) Multivariate Calculus and Differential Equations

Teaching Scheme:
Lectures : 4 hrs/week
Tutorial: 1 hr/week

Examination Scheme:
Internal Test 1 – 20 marks
Internal Test 2 – 20 marks
End Sem Exam - 60 marks

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 : Functions of several variables, level curves and level surfaces, partial and directional derivatives, differentiability, chain rule, local extreme values and saddle points. **[6 Hrs]**

Unit II : Double integrals in Cartesian and polar co-ordinates, iterated integrals, change of variables, triple integrals in Cartesian, spherical and cylindrical co-ordinates. **[11 Hrs]**

Unit III : Vector differentiation, gradient, divergence and curl, line and surface integrals, path independence, statements and illustrations of theorems of Green, Stokes and Gauss. **[10 Hrs]**

<Margin 1 inch top, bottom, left and right>

Unit IV : 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 V : Laplace Transforms, its properties, Unit step function, Dirac delta functions, Convolution Theorem, periodic functions, solving differential equations using Laplace transform. **[07 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 (12^{th} edition) by Maurice D. Weir, Joel Hass, Frank R. Giordano, Pearson Education.
- Advanced Engineering Mathematics (10^{th} 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 (4^{th} edition) by Fritz John, Springer.
- Advanced Engineering Mathematics by C.R. Wylie, McGraw Hill Publications, New Delhi.
- Advanced Engineering Mathematics (7^{th} edition) by Peter V. O' Neil, Thomson.Brooks / Cole, Singapore.
- Advanced Engineering Mathematics (2^{nd} edition) by Michael D. Greenberg, Pearson Education